Philipps-Universität Marburg
Fachbereich Mathematik und Informatik

Dissertation zur
Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat.)

Ein verteiltes Reputationssystem zur Filterung unerwünschter E-Mails

Dem Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg
vorgelegt von

Roman Meisl
aus München

Marburg/Lahn 2006

Vom Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg
als Dissertation angenommen am: 29. November 2006

Erstgutachter: Prof. Dr. Manfred Sommer
Zweitgutachter: Prof. Dr. Bernd Freisleben

Tag der mündlichen Prüfung: 14. Dezember 2006

Inhaltsverzeichnis

„Noone I know is quite sure what Spam really is, but we all agree that it is sinister-looking. The less of it there is is [in?] your life, the better.“

Chris, in: alt.angst am 25.09.1991

1 Einleitung

Täglich füllen sie zu Dutzenden die Postfächer der Mailempfänger, werben für zwielichtige Produkte oder versuchen Sicherheitslücken auf dem System des Empfängers auszunutzen. Jeder kennt sie, jeder fühlt sich durch sie gestört, und die meisten Anwender haben ihre eigene Strategie entwickelt, die Belästigung so gering wie möglich zu halten. Die Rede ist von Spam – unerwünschte, meist massenhaft verschickte E-Mails. Obwohl das Problem seit einiger Zeit existiert und sich in den letzten Jahren deutlich ausgeweitet hat, gibt es bis heute keine zufriedenstellende Lösung hierfür. Der bislang beste und am häufigsten praktizierte Ansatz ist der Einsatz sogenannter Bayesischer Spamfilter[1], welche den Inhalt einer E-Mail analysieren und anhand bedingter Wahrscheinlichkeiten berechnen, mit welcher Wahrscheinlichkeit die Nachricht unerwünscht ist.

Ist dadurch das Problem bereits gelöst? Und vor allem: Stellt Spam wirklich ein so großes Problem dar, wie von vielen Seiten immer wieder betont

[1] Siehe [86].

wird? Bayesische Spamfilter haben den Nachteil, daß sich der Empfänger niemals sicher sein kann, ob nicht doch fälschlicherweise eine wichtige Nachricht im Filter hängenbleibt oder eine Spam-Mail diesen passieren kann, die manuelle und zeitaufwendige Kontrolle kann nicht entfallen. Und daran, daß mittlerweile alle Internetprovider für ihre Kunden Anti-Spam-Module anbieten, kann bereits erahnt werden, daß Spam tatsächlich ein Problem darstellt. Letzte Zweifel sollten spätestens nach Lektüre des zweiten Kapitels ausgeräumt sein. Nach einer Definition des Begriffes „Spam“ folgen dort einige Zahlen, welche das Ausmaß belegen, gefolgt von einem Überblick über das SMTP-Protokoll und seiner den Spamversand begünstigenden Eigenschaften.

Unter diesen sticht vor allem hervor, daß der Empfänger einer Nachricht keine Möglichkeiten hat, den tatsächlichen Absender zu ermitteln. Diesem Problem widmet sich ein Großteil der in Kapitel 3 vorgestellten Methoden, darunter die beiden recht jungen, aber vielversprechenden Verfahren SPF (bzw. Sender ID) und DKIM.

Der Einsatz dieser Authentifizierungstechniken reicht jedoch nicht aus, um das Spamproblem in technischer Hinsicht zu lösen. Jedoch kann durch sie einem Absender eine gute oder schlechte Reputation zugeschrieben werden, welche wiederum als weiteres Indiz für eine Filterung der eingehenden E-Mails herangezogen werden kann. Aus diesem Grund wird in Kapitel 4 ein *verteiltes* Reputationsverfahren entwickelt, welches, im Gegensatz zu den bereits existierenden Systemen kommerzieller Anbieter, nicht auf eine zentrale Auskunftsinstanz beschränkt ist. Vielmehr ermöglicht es, durch mehrere Anfragen an sogenannte Akkreditierungspartner, eine Reputationsaussage für einen unbekannten Mailversender zu treffen. Gleichzeitig werden die gewonnen Informationen selbst für Auskünfte zur Verfügung gestellt. Dadurch steht auch Privatleuten, kleinen Unternehmen und Non-Profit-Organisationen ein Reputationsdienst zur Verfügung, für welche kostenpflichtige Angebote oft nicht in Frage kommen.

Man: *Well, what've you got?*
Waitress: *Well, there's egg and bacon; egg sausage and bacon; egg and spam; egg bacon and spam; egg bacon sausage and spam; spam bacon sausage and spam; spam egg spam spam bacon and spam; spam sausage spam spam bacon spam tomato and spam;*

Monty Python

2

Einführung in die Problematik

Zwischen „Spam? Na und, die paar Mails sind doch schnell gelöscht!“ und „Schon wieder über 100 Werbemails!“ sind alle Reaktionen bei der Betrachtung des täglichen Mail-Eingangsordners denkbar. Für den einen ist es keine große Belästigung, für den anderen eine tägliche Behinderung bei der Erledigung der anfallenden Arbeit, mit nicht unbeträchtlichem volkswirtschaftlichem Schaden, wenn man Angaben der Industrie Glauben schenken darf [31, 40].

In diesem Kapitel wird geklärt werden, daß Spam tatsächlich ein Problem ist, das einer dringenden Lösung bedarf. Hierzu wird nach einem kurzen historischen Rückblick die derzeitige Situation anhand einiger Zahlen betrachtet und es wird erläutert, welche Mechanismen dazu beitragen, daß Spam so weit verbreitet ist.

Doch zunächst ist es erforderlich den Begriff Spam enger zu fassen, zu erläutern, in welchen Bereichen er für welche Phänomene verwendet wird und was in dieser Arbeit unter Spam verstanden werden soll.

2.1 Was ist Spam?

Der Begriff Spam wird heutzutage meist mit unerwünschten, per Internet verschickten Nachrichten in Verbindung gebracht, welche oft auch nicht direkt an den Empfänger gerichtet sind. Teilweise werden inzwischen auch unerwünschte Telefonanrufe, Werbe-SMS oder Werbepost als Spam bezeichnet.

Abb. 2.1: Dose mit SPAM der Firma Hormel Foods, 1937

Der Ursprung des Wortes reicht zurück in das Jahr 1937. Seit damals vertreibt die Firma Hormel Foods unter diesem Namen ein Dosenfleisch[2] und hat den Begriff als Warenzeichen registrieren lassen

Eine zentrale Rolle spielt dieses Produkt in einem Sketch aus Monty Python's Flying Circus: In einem Restaurant fragt ein Gast nach dem Speiseangebot, worauf die Bedienung SPAM – und im Wesentlichen nur SPAM – als Gericht in den verschiedensten Variationen anpreist[3]. Auch wenn es damals von Monty Python kaum beabsichtigt war, so bringt diese Episode doch recht gut die negativen Folgen von (Internet-)Spam zum Ausdruck und könnte damit dazu beigetragen haben, daß heutzutage bei Spam nicht zuerst an Dosenfleisch gedacht wird. Wann und wie das Wort Spam erstmalig im Computerbereich verwendet wurde und inwiefern der oben erwähnte Sketch wirklich dazu beigetragen hat, läßt sich heute nicht mehr mit Gewißheit feststellen, ein recht guter Überblick über verschiedene Thesen findet sich in [94].

Bis in das Jahr 2003 hat Hormel Foods versucht, sich gegen den Mißbrauch des Wortes zu wehren, mußte dann jedoch den aussichtslosen Kampf aufgeben[4].

---

[2] „Spiced Ham", also gewürzter Schinken; fälschlicherweise oft auch als Akronym für „**s**piced **p**ork **a**nd **m**eat/ha**m**" bzw. „**s**houlder **p**ork **a**nd **m**eat".

[3] Siehe z. B. [65].

[4] [46]. Gemäß dieser Erklärung wird in dieser Arbeit das Dosenfleisch mit SPAM und eine unerwünschte E-Mail mit Spam bezeichnet.

2.1.1 Analyse der verschiedenen Spamarten

Als Spam werden heutzutage teilweise sehr unterschiedliche „Nachrichten“, welche über verschiedene Kanäle verbreitet werden, bezeichnet. Aus diesem Grund ist eine genauere Betrachtung angebracht, was mit Spam bezeichnet wird. Hierbei kann einerseits nach Inhalt und andererseits nach Medium differenziert werden.

2.1.1.1 Klassifizierung nach Medium

Natürlich können auch unerwünschte Nachrichten abseits des Computers als Spam bezeichnet werden, wie beispielsweise Postwurfsendungen. Besonders interessant ist hierbei die Entwicklung auf dem Telefonsektor. Vor der Liberalisierung dieses Marktes waren Anrufe für Werbung, Umfragen oder Gewinnspiele recht selten. Nachdem das Monopol der Telekom aufgehoben wurde und infolgedessen die Verbindungskosten deutlich gefallen sind, hat die Anzahl solcher Anrufe deutlich zugenommen, obwohl sie rechtlich unzulässig sind.

Der Begriff Spam taucht im Zusammenhang mit folgenden unterschiedlichen Bereichen des Internet auf:

- **WWW:** Mit Suchmaschinen-Spam werden meist spezielle HTML-Seiten bezeichnet, mit denen erreicht werden soll, daß eine bestimmte Internetseite oder eine ganze Site bei den Suchergebnissen möglichst weit oben plaziert ist. Diese Art von Spam spielt für diese Arbeit keine Rolle und wird nicht weiter betrachtet. Darüber hinaus besteht das Problem, daß von Spammern programmierte Suchmaschinen das WWW nach validen Mailadressen durchsuchen.

- **Usenet:** Spam im Usenet zeichnet sich durch sehr häufige Postings in eine oder mehrere Newsgroups aus. Hierbei kann wie folgt unterschieden werden:

  - die gleiche Nachricht wird an sehr viele (oder sogar alle) Newsgroups verschickt

  - eine oder wenige Gruppen erhalten sich sehr oft wiederholende oder stark ähnelnde Postings
  - Mischformen der beiden genannten Möglichkeiten[5]

  Da die meisten Newsserver mit einem Zugangsschutz (Filterung auf IP-Ebene, Abfrage von Zugangsdaten etc.) ausgestattet sind und meist über ein funktionierendes Abusemanagement verfügen, ist die Spamproblematik zwar gegeben, aber bei weitem nicht so stark ausgeprägt wie bei dem Medium E-Mail und soll hier ebenfalls nicht weiter diskutiert werden. Analog zum WWW existiert das Problem der Sammlung von E-Mail-Adressen durch Agenten.

- **E-Mail:** Per E-Mail verbreiteter Spam ist dem im Usenet sehr ähnlich, jedoch sind die Hürden für den Absender leichter zu überwinden. Wie weiter unten dargelegt wird, bietet SMTP so gut wie keinen Schutz gegen Spam. Mögliche Empfängeradressen werden – wie bereits oben erwähnt – durch spezialisierte Suchmaschinen automatisiert im WWW und im Usenet gesammelt, von Adreßhändlern verkauft, aus Kundendatenbanken extrahiert oder einfach durch bloßes Ausprobieren gewonnen. Da die Spamproblematik im Zusammenhang mit E-Mail am stärksten ausgeprägt zu sein scheint, ist dies der Schwerpunkt dieser Arbeit.

2.1.1.2 Klassifizierung nach Inhalt

Von seiten des Benutzers werden inhaltlich teils sehr unterschiedliche Mails als Spam bewertet, auch solche, welche eigentlich nicht unbedingt als Spam zu sehen sind. Die Bandbreite reicht hierbei von Massenmails mit anstößigem Inhalt über Viren oder Würmer bis hin zu Mails, welche der Benutzer in Form eines Newsletter womöglich vor längerer Zeit sogar explizit angefordert hat. Hierbei hilft folgende Kategorisierung, welche leider nicht immer eine scharfe Trennung möglich macht:

[5] Zur besseren Einschätzung, ob ein Posting als Spam zu werten ist, kann der sog. Breidbart-Index herangezogen werden [102].

UBE – Unsolicited Bulk E-Mail Als Unsolicited Bulk E-Mail, also unerwünschten Massenmails, werden E-Mails bezeichnet, welche an Hunderttausende oder gar Millionen Empfänger gleichzeitig gerichtet sind. Typischerweise besteht zwischen Absender und Empfänger keine wie auch immer geartete Beziehung, und falls ein Produkt beworben wird, darf sein Nutzen oder seine Wirkung stets in Zweifel gezogen werden. Hierunter fallen z. B. Werbung für Vergrößerung von Körperteilen, Beschwerdemails als Antwort an die (meist gefälschte) Absenderadresse etc. Wird in der E-Mail eine Gelegenheit zum leichten Geldverdienen angepriesen, spricht man auch von „Scam“ (Nigeria-Connection). Wird als Absender einer UBE ein unbeteiligter Dritter angegeben mit dem Ziel, diesem Schaden zuzufügen, wird dies als „Joe-Job“ bezeichnet.

UCE – Unsolicited Commercial E-Mail Als Unterform der UBE wird die Unsolicited Commercial E-Mail, die unerwünschte kommerzielle E-Mail betrachtet, mittels der für ein reales Produkt geworben wird. Die Empfängeradressen stammen meist aus Kundendatenbanken, und eine Einwilligung in den Empfang solcher Werbung wird oft durch nebulöse AGBen oder unübersichtliche Registrierungsformulare erlangt.

Viren & Würmer Zunehmend nutzen auch Autoren von Viren und Würmern das Medium E-Mail, um damit ihre Schadsoftware zu verbreiten. Solche Mails sind meist mit einem verlockenden Attachement versehen oder mit einem Link auf eine vielversprechende Homepage. Ziel des Absenders ist dabei immer, eine Sicherheitslücke der Software des Empfängers auszunutzen.

Phishing Unter Phishing versteht man den *groß angelegten* Versuch, Zugangsdaten zu erschleichen, also letztendlich einen Identitätsdiebstahl. Meist wird hierzu massenhaft E-Mail verschickt, welche mittels HTML das Design einer großen Firma täuschend echt imitiert. Unter einem Vorwand (Überprüfung der Zugangsdaten, Beschädigung der Datenbank etc.) wird der Empfänger dazu aufgefordert, seine Zugangsdaten mitzuteilen, entweder per Antwortmail oder indem er sich auf einer

imitierten Homepage anmeldet.

Hoax Der klassische Hoax ist eine per E-Mail verbreitete Warnung vor einem angeblichen Virus, welcher die Aufforderung beigefügt ist, diese Warnung an möglichst alle Freunde und Bekannte weiterzusenden. Mittlerweile werden so auch Ketten-E-Mails bezeichnet, welche auf ein bestimmtes Ereignis aufmerksam machen wollen und sich nach dem Schneeball-Prinzip verbreiten.

Normale E-Mail Fälschlicherweise wird oft auch eine normale E-Mail vom Empfänger als Spam interpretiert, z. B. Hinweise auf Updates einer direkt beim Hersteller gekauften Software, Sicherheitshinweise oder E-Mails einer Mailingliste mit sehr wenig Verkehr, auf der sich der Empfänger vor längerer Zeit eingetragen hat. Dies erweist sich vor allem im Zusammenspiel mit Bayesischen Spamfiltern als problematisch.

2.1.2 Definition des Begriffes Spam

Wie im vorigen Abschnitt deutlich geworden ist, wird der Begriff Spam sehr vielfältig und in verschiedenen Zusammenhängen verwendet. Dementsprechend gibt es zahlreiche Versuche, den Begriff Spam genauer zu definieren. In der Rechtssprechung kann eine Definition für Spam dem § 7 des Gesetzes gegen den unlauteren Wettbewerb (UWG) entnommen werden. Demnach sind Nachrichten, welche folgende Merkmale aufweisen, in Deutschland nicht zulässig:

1. Die Nachricht muß werbenden Charakter haben.
2. Die Nachricht ist unerwünscht, bzw. eine Einwilligung des Empfängers liegt nicht vor.
3. Zwischen Sender und Empfänger besteht kein geschäftlicher Kontakt.
4. Die Identität des Absenders wird verschleiert oder verheimlicht, bzw. die Nachricht enthält keine gültige Adresse, an die eine Aufforderung zur Einstellung der Zusendungen geschickt werden könnte.

Die Anti-Spam-Organisation „Spamhaus“ definiert Spam als unerwünschte Massenmail: „A message is Spam only if it is both Unsolicited *and* Bulk.“ [91] Unerwünscht (unsolicited) bedeutet, der Empfänger hat keine überprüfbare Zustimmung für das Zusenden der Nachricht gegeben. Massenhaft (bulk) bedeutet, daß eine größere Anzahl Empfänger die gleiche Nachricht erhalten haben[6].

Der Internetprovider WEB.DE überläßt die Definition seinen Nutzern: „Spam ist das, was der User als Spam markiert.“[7]

Eine der Kommunikationstheorie entsprechende Definition könnte wie folgt formuliert werden: Spam ist die deutliche Belastung eines Nachrichtenkanals mit unerwünschten und/oder unaufgeforderten, aber protokollkonformen Nachrichten, welche die Kommunikation über diesen Kanal erheblich erschwert oder unmöglich macht. Dabei ist die Störung nicht Ziel sondern Nebeneffekt.

Im Rahmen dieser Arbeit ist Spam eine per E-Mail zugestellte Nachricht, deren Zusendung nicht erwünscht ist und vom Empfänger nicht verlangt wurde und dadurch eine Belästigung darstellt oder die E-Mail-Kommunikation erschwert.

2.2 Das Phänomen Spam

Der folgende Abschnitt erläutert, welches Ausmaß Spam in den letzten 30 Jahren erreicht hat, und macht klar, daß der massenhafte Versand von Werbemails nicht nur ein Ärgernis ist, welches subjektiv unterschiedlich wahrgenommen wird. Die ganze Dimension des Problems wird erst bei der Betrachtung verschiedener, von Providern und Internet-Analysten bereitgestellter Zahlen deutlich.

---

[6] Gleich ist hier zu verstehen als: Die Identität des Empfängers ist unerheblich, da die Nachricht genauso gut an andere Empfänger gerichtet sein könnte. (D. h. abgesehen vom Header bestehen Unterschiede lediglich in Begrüßungsformeln etc.)

[7] Leslie Romeo, Projektleiter Spam- und Virenschutz, WEB.DE AG auf dem 2. Deutschen Anti Spam Kongress in Köln, 22. September 2004.

2.2.1 Ein kurzer historischer Abriß

Als erste Spam-Nachricht wird häufig die Werbung des Rechtsanwaltes Lawrence Canter angesehen, mit welcher er ab Februar 1994 in mehreren Newsgroups[8] Werbung für seine Kanzlei zum Thema „US Green Card Lottery“ machte[9].

Tatsächlich muß wohl aber eine Massenmail aus den Zeiten des Arpanet als erster – wenn auch dilettantischer – Spamversuch gewertet werden [95]. Im Jahr 1978 bewarb Gary Thuerk, ein Angestellter der Firma DEC, die neue DECSYSTEM-20-Familie, deren Betriebssystem als neueste Errungenschaft Unterstützung der Arpanet-Protokolle bot.

Auch wenn der Cantersche Versuch nicht der erste gewesen sein mag und diese Werbung nicht per E-Mail verbreitet, sondern im Usenet gepostet wurde, so macht folgendes diese Nachricht rückblickend bemerkenswert:

- Die Nachricht wurde mittels eines Perlscripts *gleichzeitig* an *alle* Newsgroups verschickt.
- Die *Einnahmen* seiner Kanzlei sollen durch diese Aktion um mehrere tausend Dollar gestiegen sein.
- Der Effekt war nicht nur ein erhöhter Umsatz, sondern Lawrence Canter trat damit eine Lawine an Beschwerde-Postings los, weil sich zahlreiche Nutzer des Usenet *gestört* fühlten.
- In den Newsgroups wurde ausführlich darüber diskutiert, inwiefern Werbung im Usenet erlaubt sein soll. Im Rahmen dieser Diskussionen wurde sehr häufig das Wort Spam verwendet und somit als Begriff für unerwünschte, werbende Massenmails geformt.

Ab dem Jahr 1994 fand massenhafte Werbung im Internet zunehmend Verbreitung, und zwar sowohl per Mail als auch im Usenet.

[8] Zunächst soc.culture.* u. a., später alle Newsgroups.
[9] Siehe [22].

2.2.2 Das Ausmaß des Problems

In den letzten Monaten und Jahren hat die Diskussion um Spam verstärkt Einzug in die Politik gehalten und sowohl auf nationaler als auch auf internationaler Ebene zu verschiedenen Gesetzesinitiativen geführt. So trat beispielsweise in den USA am 1.1.2004 der CAN-SPAM Act [73] in Kraft, und in Australien wurde 2003 der Spam Act verabschiedet. Am 21.1.2002 wurde die EU-Datenschutzrichtlinie 2002/58/EC [36] verabschiedet, wonach das Versenden unverlangter E-Mails verboten ist. Da diese Richtlinie derzeit noch nicht in hiesiges Recht umgesetzt ist, ist in Deutschland die einzige juristische Handhabe gegen Spammer das Gesetz gegen den unlauteren Wettbewerb (UWG), wonach jedoch nur Mitbewerber, Verbraucherschutzverbände und Wettbewerbszentralen, nicht jedoch Privatleute klageberechtigt sind.[10]

Festzustellen ist, daß sich in den letzten Jahren der aus den USA stammende Spam-Anteil stetig verringert hat, von 56,7% (Februar 2004) auf 23,1% (April 2006) [89, 90]. Da absolute Zahlen fehlen, kann nur gemutmaßt werden, ob dies auf den CAN-SPAM Act zurückzuführen ist, oder ob die Zahlen möglicherweise durch eine Abwanderung aus den USA oder durch eine stärkere Zunahme in anderen Ländern zu erklären sind.

Obwohl aufgrund dieser Gesetze bereits erste Urteile mit Haftstrafen und teilweise empfindlichen Geldstrafen – mehrere Millionen bis Milliarden US-Dollar[11] – gefällt wurden, scheint die juristische Handhabe – sofern sie tatsächlich wirksam sein sollte – bislang nicht auszureichen, wie nachfolgende Zahlen belegen.

	Anzahl	Mails/Tag	Prozent
Normale Mails	2 510	13	14,80 %
von Mailinglisten	8 059	44	47,50 %
Spam-Mails	6 395	34	37,70 %
gesamt	16 964	91	100,00 %

Tab. 2.1: Mailstatistik (15.8.2004 bis 15.2.2005)

[10] Einen guten Überblick über die Bestimmungen und Gesetze in verschiedenen Ländern gibt [75].

[11] Siehe [30, 34–36, 38, 41, 42, 45].

	Anzahl	Mails/Tag	Prozent
Normale Mails	2 770	15	6,84 %
von Mailinglisten	13 842	75	34,17 %
Spam-Mails	23 897	129	58,99 %
gesamt	40 509	219	100,00 %

Tab. 2.2: Mailstatistik (20.1.2006 bis 24.7.2006)

Über zwei jeweils 185 Tage dauernde Zeiträume wurden die eingehenden Nachrichten mehrerer E-Mail-Adressen analysiert und in die Kategorien „Normale Mail“, „Mailingliste“ und „Spam“ eingeteilt (siehe Tabellen 2.1 und 2.2). Die erwähnten Mailinglisten enthalten vor allem den Mailverkehr von stark frequentierten Listen wie Bugtraq und Full Disclosure. Würde dieser nicht mitgerechnet ergäbe sich sogar eine Spam-Quote von 60,7% (bzw. 88,4%). Einen besseren Überblick verschafft die Betrachtung folgender Zahlen und Meldungen:

- Auf dem 2. Deutschen Anti Spam Kongress wurden von WEB.DE folgende Zahlen veröffentlicht:

	Spam (Prozent)	Rejects
web.de – Normal	15 Mio (60%)	15 Mio
web.de – Spitzenzeiten	30 Mio (85%)	35 Mio

Tab. 2.3: Spamstatistik von WEB.DE: Durchschnittliche Anzahl von Spam-Mails pro Tag. Stand: September 2004

- Bei T-Online werden jeden Tag (April 2006) ca. 1 Milliarde Spam-Mails gefiltert. Dem gegenüber stehen ca. 30 Millionen normale E-Mails [43].

- 2004 überstieg der Spamanteil in Deutschland erstmalig die 40%-Marke [33].

- Der durchschnittliche globale Spam-Anteil lag laut MessageLabs 2004 bei 73,2% und 2005 bei 68,6% [59, 60].

- Jeden Tag (Mai 2005) entstehen ca. 172 000 neue Spam-Relays [37].

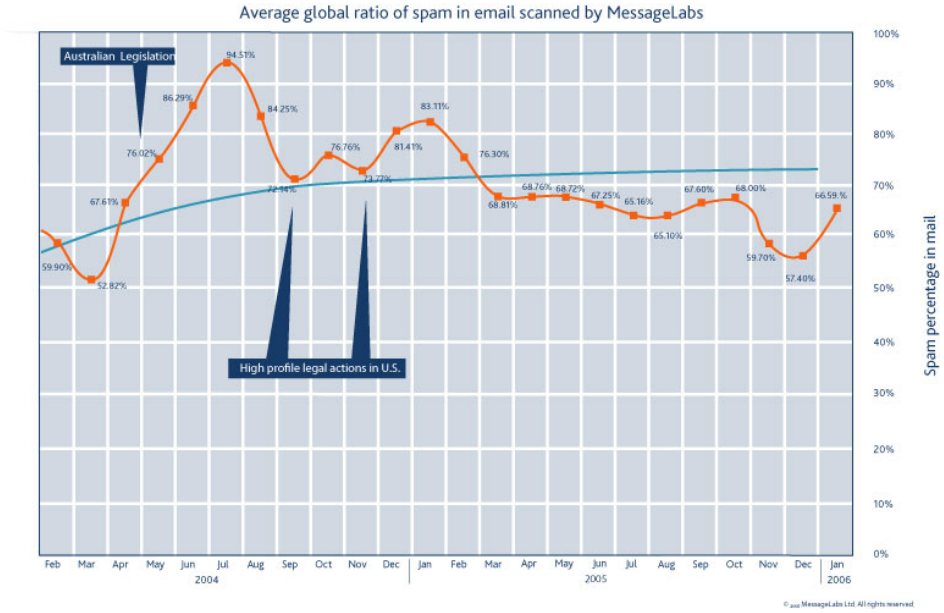

Abb. 2.2: Durchschnittlicher globaler Spam-Anteil der letzten Jahre laut MessageLabs [58]

Auch wenn der Spam-Anteil in den letzten Jahren stagniert oder sogar leicht rückläufig ist (siehe Abbildung 2.2), sind die Zahlen alarmierend genug, um dringend Maßnahmen zur Eindämmung der Flut von unerwünschten Nachrichten zu ergreifen, zumal von den eingehenden Spam-Mails ein nicht zu vernachlässigendes Sicherheitsrisiko ausgeht, wie im nachfolgenden Abschnitt erläutert wird.

2.2.3 Gefahrenpotential

Abgesehen von dem beim Empfänger entstehenden und wohl eher zu vernachlässigenden Schaden durch erhöhte Downloadkosten birgt ein hohes Spamaufkommen natürlich das Risiko, daß wichtige Nachrichten übersehen oder irrtümlich gelöscht werden.

Spätestens seitdem regelmäßig Viren und Würmer wie Loveletter, Blaster oder MyDoom für Spamwellen sorgen, wird klar, daß Spam nicht nur eine erhebliche Belästigung, sondern auch ein konkretes Gefahrenpotential mit sich bringt, welches nicht ignoriert werden darf. Hierbei kann unterschieden

werden zwischen Würmern, welche per Mailanhang die Schwachstellen des Mailprogramms des Empfängers ausnutzen, und Viren und Würmern, welche sich anderweitig Zugang zu einem System verschaffen, um dort ein eigenes SMTP-Modul zu installieren, welches unter anderem der eigenen Weiterverbreitung und dem Versand von Spam dient[12].

Die zweite deutliche Gefahr liegt in den schon weiter oben erwähnten Phishing-Mails, mit denen versucht wird, vertrauliche Daten und Anmeldeinformationen zu erschleichen. Hieraus kann den Betroffenen erheblicher finanzieller Schaden entstehen, wenn der Angreifer auf diesem Wege z. B. Zugangsdaten zu einem Bankkonto oder einem eBay-Account erlangt.

Sowohl gegen die Bedrohung durch Malware (Viren und Würmer) als auch gegen Phishing-Versuche kann auf Protokollebene wenig unternommen werden. Denn die zur Zeit per E-Mail verbreitete Schadsoftware nutzt entsprechende Protokolle nur zur eigenen Weiterverbreitung, nicht aber Sicherheitslücken *in* den Protokollen, um Zugriff auf ein System zu erhalten. Letztendlich sind es gutgläubige Nutzer, welche auf die angegebenen Absenderinformationen vertrauen und entsprechende Dateien bedenkenlos öffnen und somit die Schadsoftware selbst zu Ausführung bringen. Sobald ein Kommunikationsmittel den Austausch von Dateien ermöglicht, ist also auch die Möglichkeit gegeben, darüber Viren oder Würmer zu verbreiten. Der einzige Ausweg besteht darin, dem Empfänger Gewißheit über die Identität des Absenders zu geben und damit die selbsttätige Weiterverbreitung zu erschweren. Dies könnte beispielsweise durch vorhandene Authentifizierungsmerkmale in E-Mails geschehen.

Insgesamt dürfte das unmittelbar von Spam ausgehende Gefahrenpotential als eher gering einzuschätzen sein und vor allem sehr stark von den individuellen Fähigkeiten der Nutzer und der Konfiguration der eingesetzten Software abhängen. Festzuhalten ist jedoch, daß durch Spam das Vertrauen in die E-Mail-Kommunikation geschwächt wird, das manuelle Ausfiltern als ungemein lästig empfunden wird und die Begleiterscheinungen teilweise zu erheblichen Kosten führen [32, 40].

[12] Siehe auch [2].

2.3 Das Simple Mail Transfer Protocol

Die Kommunikation per E-Mail wird als eine der wichtigsten Anwendungen des Internet gesehen. Vor allem im gewerblichen Umfeld ist sie unersetzlich geworden und hat sich neben anderen Kommunikationsverfahren wie Telefon, Telefax und Brief längst etabliert. Die entscheidende Rolle hierbei spielt das Simple Mail Transfer Protocol, kurz SMTP, welches den Versand und Transport der Nachricht regelt.

Wie bei vielen Protokollen aus den frühen Tagen des Internet wurde bei der Entwicklung weniger auf Sicherheit, sondern mehr auf Stabilität und Zuverlässigkeit geachtet. Als 1982 SMTP mit dem RFC 0821 zum Standard erklärt wurde, konnte sich wohl einfach niemand vorstellen, daß eines Tages massenhaft Mails verschickt und Absenderadressen gefälscht werden. Nur so ist zu erklären, daß es durch SMTP so leicht ist, Spam zu versenden. In den nun folgenden Abschnitten werden die Schwachpunkte von SMTP genauer dargelegt.

2.3.1 Eine kurze Einführung

Die aktuelle Fassung des Simple Mail Transfer Protocol wird im RFC 2821 beschrieben und stammt vom April 2001. Von Anfang an stand die Stabilität des Protokolls im Vordergrund; das zentrale Ziel ist die zuverlässige und leistungsfähige Übertragung von E-Mails. SMTP ist dabei nicht vom zugrunde liegenden Netzwerk abhängig, solange dieses einen gesicherten Transport bereitstellt, wie beispielsweise TCP. SMTP ist damit im OSI-Schichtenmodell oberhalb der Transportschicht anzusiedeln, vor allem in den Schichten 5 (Sitzungsschicht) und 6 (Darstellungsschicht). Eine wichtige Eigenschaft von SMTP ist dabei das sogenannte Mail-Relaying, der Transport von Nachrichten über verschiedene Netzwerke. Dadurch kann die Nachricht nicht nur auf direktem Weg zum Empfänger gelangen, sondern auch über mehrere Zwischenstationen, und das System kann flexibel auf Ausfälle und Unterbrechungen reagieren. Eine wichtige Rolle hierbei spielt der Mail eXchanger-Mechanismus des Domain Name System, welcher in Kapitel 2.3.2 näher erläutert wird.

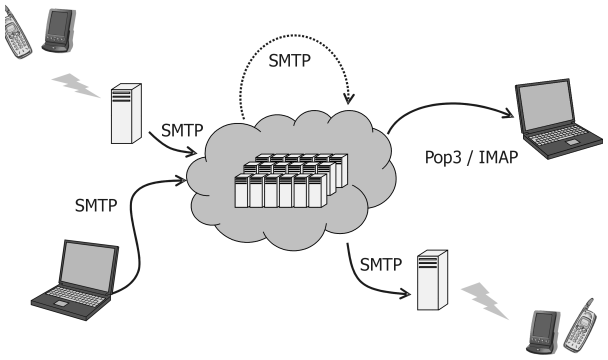

Abb. 2.3: Mögliche Wege einer E-Mail vom Sender zum Empfänger

Die Kommunikation vom Absender einer E-Mail bis zum Empfänger stellt sich in der Regel wie folgt dar: Der Absender schickt die Nachricht mittels SMTP an einen Server (oft ein Mailserver eines Internetproviders), welcher sich um den Weitertransport kümmert. Die weitere Übertragung – eventuell über weitere SMTP-Server anderer Netzbetreiber – erfolgt ebenfalls per SMTP bis zum Mailserver des Empfängers, bei dem die Nachricht zur Abholung durch den Empfänger mittels anderer Protokolle (z. B. POP3 oder IMAP) bereitliegt. Die Abbildung 2.3 veranschaulicht die möglichen Wege.

Durch moderne, tragbare Geräte wie Mobiltelefon oder PDA (sog. Mobile Devices) ist der Versand und Empfang von E-Mails möglich, ohne direkt an das Internet angeschlossen zu sein. Insgesamt wird laut RFC 2821[13] zwischen vier verschiedenen Typen von SMTP-Systemen unterschieden:

1. Das Ursprungssystem, von welchem die Nachricht in das Transportsystem eingebracht wurde.

2. Das Zielsystem, auch Mail Delivery Agent (MDA) genannt, von welchem die Nachricht direkt an den Empfänger gesendet oder durch diesen von dort abgeholt werden kann.

3. Eine Zwischenstation, oder Relay, welche zunächst als SMTP-Server agiert und die Nachricht von einem SMTP-Client entgegennimmt, anschließend selbst als SMTP-Client tätig wird und sie ohne Modifizie-

[13] Siehe [52] S.12.

rung außer dem Hinzufügen von Trace-Informationen (Received-Header) an einen SMTP-Server weiterreicht.

4. Ein Gateway, welches eine Nachricht aus dem einen Transportsystem empfängt und an einen Server in einem anderen Transportsystem weiterreicht, wobei eines der beteiligten Transportsysteme SMTP verwendet.

Die Endpunkte der Kommunikationskette, bei denen die Nachrichten in das Transportsystem eingespeist bzw. daraus entfernt werden (also die Mailprogramme der Benutzer), werden als Mail User Agent (MUA) bezeichnet. Sie verfügen in der Regel nur über eine eingeschränkte SMTP-Funktionalität.

Server, die für die Annahme einer E-Mail per SMTP zuständig sind, werden als Mail Transfer Agent (MTA) bezeichnet und müssen über die volle SMTP-Funktionalität verfügen.

Eine typische Übertragung einer Mail von Server zu Server per SMTP spielt sich dabei immer wie folgt in einem Request-Response-Zyklus ab:

```
220 pc12717.mathematik.uni-marburg.de ESMTP server ready.
EHLO moutng.kundenserver.de
250-pc12717.mathematik.uni-marburg.de Hello moutng.kundenserver.de; ¶
                          ESMTPs are:
250-TIME
250-SIZE 0
250 HELP
MAIL FROM:<user@example.de> SIZE=874
250 Sender and size (874) OK - send RCPTs.
RCPT TO:<user1@pc12717.mathematik.uni-marburg.de>
250 Recipient OK - send RCPT or DATA.
DATA
354 OK, send data, end with CRLF.CRLF
Received: from [137.248.121.158] (helo=pc12717.Mathematik.Uni-Marburg.DE)
  by mrelayeu.kundenserver.de (node=mrelayeu1) with ESMTP (Nemesis),
  id 0MKwpI-1G1J2q3Xh7-00006s; Fri, 14 Jul 2006 10:29:08 +0200
Date: Fri, 14 Jul 2006 10:29:08 +0200
From: Roman Meisl <user@example.de>
Reply-To: Roman Meisl <user@example.de>
X-Priority: 3 (Normal)
Message-ID: <1384690538.20060714102908@example.de>
To: user1@pc12717.mathematik.uni-marburg.de
Subject: SMTP-Veranschaulichung
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-Provags-ID: kundenserver.de abuse@kundenserver.de
login:9cfad2e2cf66172b135409a015bd7214

Hier beginnt der Text...
```

```
....und hier endet er
31 --
'Multiple exclamation marks,' he went on, shaking his head, 'are a
33 sure sign of a diseased mind.' (Terry Pratchett)
.
35 250 Data received OK.
QUIT
37 221 pc12717.mathematik.uni-marburg.de Service closing channel.
```

Codebeispiel 2.1: Ein typischer Nachrichtentransfer per SMTP

Zeilen, die mit einem dreistelligem Zahlencode beginnen (z. B. Zeilen 1, 3-6, 8 usw.), sind Antworten des empfangenden Servers, Zeilen ohne Zahlencode sind Befehle des Senders (z. B. Zeilen 2, 7, 9 usw.). Die Zeilen 13 bis 33 enthalten die eigentliche E-Mail, welche sich sich in den Header (Zeilen 13 bis 27) und den Body (Zeilen 29 bis 33) untergliedert. Beide Teile sind durch eine Leerzeile (Zeile 28) getrennt.

Im folgenden Abschnitt werden die Begriffe Sender oder Client für den Computer verwendet, welcher eine E-Mail übertragen möchte und somit die Kommunikation initiiert. Mit Empfänger oder Server wird der Computer bezeichnet, welcher die E-Mail empfängt (und für den Adressaten zwischenspeichert oder in einem späteren, von der aktuellen Übertragung unabhängigen Schritt an einen anderen Server weiterleitet[14]). In der folgenden Ausführung wird eine typische SMTP-Sitzung mit den wichtigsten und häufigsten Abweichungen beschrieben. Für einen vollständigen Überblick wird auf das RFC 2821 verwiesen.

Eine SMTP-Sitzung beginnt mit der Herstellung einer Verbindung vom Client zum Server, worauf dieser mit einer Statusmeldung (z. B. Rechnername, Serversoftware und -version, Datum und Uhrzeit) antwortet. Üblicherweise wird die Sitzung durch den Client mit dem `EHLO`-Befehl eröffnet, welcher darüber hinaus dem Server mitteilt, daß sog. *Service Extensions* unterstützt werden. Dieser Befehl sollte, muß aber nicht von beiden Seiten unterstützt werden, statt dessen kann auch das einfachere `HELO` verwendet werden.

Mit dem Befehl `MAIL FROM:<reverse-path>`, welcher die Absenderadresse enthält, wird die Übertragung der E-Mail eingeleitet. Daraufhin folgt einer

[14] In diesem Fall würde dann dieser Computer von der Rolle des Servers in die Rolle des Clients wechseln.

oder mehrere `RCPT TO:<forward-path>`-Befehle, anhand dessen der oder die Empfänger der E-Mail spezifiziert werden. Nachdem der Sender den `DATA`-Befehl gesendet und der Empfänger diesen quittiert hat, kann der Transfer der Maildaten erfolgen. Mit einer Zeile, welche nur einen Punkt enthält wird die Übertragung beendet, und der Server kann die Maildaten und die Empfängeradressen verarbeiten, was von diesem mit einer `250 Data received OK`-Meldung bestätigt wird.

An dieser Stelle kann entweder mit einem weiteren `MAIL FROM`-Befehl die Übertragung einer neuen E-Mail begonnen werden oder mittels `RSET` (Zurücksetzen des Servers) und `QUIT` die Sitzung beendet werden.

Die Antworten des Servers bestehen aus einem dreistelligen Statuscode gefolgt von einem erläuternden Text. Für die kommunizierenden Programme ist allein der Zahlencode ausschlaggebend, der Text dient nur der besseren Verständlichkeit und kann bei identischem Statuscode je nach Software variieren. Die Statusmeldungen lassen sich, wie in Tabelle 2.4 dargestellt, in fünf Gruppen einteilen.

Code	Bedeutung
1yz	Vorläufige Bestätigung; *der Befehl wurde akzeptiert, benötigt aber noch eine Bestätigung der in dieser Antwort enthaltenen Daten. (Kann nur in Verbindung mit Service Extensions vorkommen.)*
2yz	Bestätigung; *der Befehl wurde akzeptiert und verarbeitet. Weitere Befehle werden erwartet.*
3yz	Einstweilige Bestätigung; *der Befehl wurde akzeptiert, bedarf aber weiterer Daten, welche in einem weiteren Befehl gesendet werden müssen.*
4yz	Vorübergehender Fehler; *der Befehl wurde aufgrund eines vorübergehenden Fehlers nicht akzeptiert. Eine spätere Wiederholung ist möglich, der Sender sollte mit der Befehlssequenz neu beginnen.*
5yz	Permanenter Fehler; *der Befehl wurde aufgrund eines Fehlers nicht akzeptiert, der Sender sollte diesen Befehl während dieses Vorgangs nicht noch einmal senden.*

Tab. 2.4: SMTP Statuscodes

Die nach dem `DATA`-Befehl übermittelten Daten umfassen die eigentliche

E-Mail, bestehend aus Header und Body. Diese Daten bleiben unverändert bis auf die Tatsache, daß am Anfang der Headerzeilen eine Received-Headerzeile eingefügt wird, welche darüber Auskunft gibt, von welchem Host die Mail empfangen und von welchem sie gesendet wurde. Anhand dieser Headerzeilen – sofern sie nicht gefälscht sind – kann nachvollzogen werden, welchen Weg die E-Mail genommen hat.

Die restlichen während der SMTP-Sitzung ausgetauschten Daten, vor allem die Mailadressen aus dem Reverse-path und dem Forward-path, bilden den sogenannten *Envelope*, welcher bei jeder SMTP-Sitzung neu erstellt wird.

Hat eine Nachricht mehrere Empfänger (mehrere `RCPT TO`-Befehle), wird für alle Empfänger am gleichen Bestimmungsort nur eine Kopie verschickt. Folgendes Beispiel veranschaulicht diesen Vorgang; die SMTP-Sitzungen sind auf den relevanten Teil gekürzt.

```
...
MAIL FROM:<sender@domain.tld>
250 2.1.0 <sender@domain.tld>... Sender ok
RCPT TO:<albert@destination.tld>
250 2.1.5 <albert@destination.tld>... Recipient ok
RCPT TO:<berta@destination.tld>
250 2.1.5 <berta@destination.tld>... Recipient ok
RCPT TO:<hans@another-dest.tld>
250 2.1.5 <hans@another-dest.tld>... Recipient ok
RCPT TO:<wurst@another-dest.tld>
250 2.1.5 <wurst@another-dest.tld>... Recipient ok
DATA
...
```

Codebeispiel 2.2: Nachrichtentransfer mit mehreren Empfängern, Eingang am SMTP-Server

Diese Nachricht hat zwei Empfänger in der Domain `destination.tld` und zwei in der Domain `another-dest.tld`. Der Versand dieser Nachricht erfolgt in zwei getrennten Sessions. Zunächst der Versand der Nachricht nach `destination.tld`:

```
...
MAIL FROM:<sender@domain.tld>
250 mail from: <sender@domain.tld> ok
RCPT TO:<albert@destination.tld>
250 <albert@destination.tld> ok
RCPT TO:<berta@destination.tld>
250 <berta@destination.tld> ok
DATA
...
```

Codebeispiel 2.3: Nachrichtentransfer mit mehreren Empfängern, Ausgang, Session 1

Und anschließend *eine* Kopie nach `another-dest.tld`:

```
...
MAIL FROM:<sender@domain.tld>
250 2.1.0 <sender@domain.tld>... Sender ok
RCPT TO:<hans@another-dest.tld>
250 2.1.5 <hans@another-dest.tld>... Recipient ok
RCPT TO:<wurst@another-dest.tld>
250 2.1.5 <wurst@another-dest.tld>... Recipient ok
DATA
...
```

Codebeispiel 2.4: Nachrichtentransfer mit mehreren Empfängern, Ausgang, Session 2

Das heißt, werden Kopien einer Nachricht benötigt, werden sie auf dem Verbreitungsweg so spät wie möglich erzeugt.

2.3.2 Das Domain Name System im Zusammenhang mit SMTP

Das Domain Name System (DNS) hat neben der üblichen Namensauflösung beim Versand von E-Mails eine weitere Aufgabe. Wie in den RFCs 974, 1034, 1035 und 2821 beschrieben[15], dient es auch als Verzeichnis für den oder die Maileingangsserver einer Domain. Hierfür steht ein gesonderter Resource Record zur Verfügung (MX-RR, Mail eXchange RR), anhand dessen die Mailserver spezifiziert werden können:

```
DOMAIN.TLD. MX   10 SRV1.DOMAIN.TLD.
            MX   10 SRV2.DOMAIN.TLD.
```

Codebeispiel 2.5: Beispiel eines MX-RR

Diese Konfiguration besagt, daß die Hosts `SRV1` und `SRV2` für den Empfang von Mails an die Domäne `DOMAIN.TLD`. gleichberechtigt zuständig sind. Der Zahlenwert vor dem Hostnamen (hier jeweils 10) drückt die Präferenz aus. Er ist dabei als „Entfernung“ zu verstehen; je kleiner er ist, desto größer ist die Präferenz.

Sendende SMTP-Server sind verpflichtet, eine DNS-Abfrage nach einem MX-RR zu stellen, um das Ziel der Nachricht zu erhalten. Hierzu wird aus der Mailadresse des `RCPT TO`-Befehls der Domainname extrahiert und eine entsprechende Anfrage an den in der Konfiguration des SMTP-Servers eingetragenen Nameserver gerichtet. Im Normalfall antwortet dieser mit einem

[15] [52, 63, 64, 76].

oder mehreren MX-Records. Daraus erstellt der sendende SMTP-Server eine nach aufsteigendem Präferenzwert sortierte Liste und versucht, beginnend mit dem ersten Mailserver dieser Liste, die Nachricht zu verschicken. Im Fall eines erfolglosen Sendeversuches wird der nächste Server der Liste (mit möglicherweise höherem Präferenzwert) ausgewählt und ein neuer Sendeversuch unternommen.

2.3.3 Schwächen von SMTP

Folgende Eigenschaften des SMTP-Protokolls erschweren es dem Empfänger, sich gegen unerwünschte Nachrichten zu wehren, bzw. erleichtern den Versand von Spam:

Vielfältige Absenderangaben In den Headerzeilen einer E-Mail können insgesamt fünf verschiedene Absenderadressen stehen:

1. Der `From`-Header bezeichnet den Autor einer Nachricht, also die Adresse jener Person oder jenes Systems, welches den Inhalt der Nachricht erstellt hat.

2. Der `Sender`-Header bezeichnet die Adresse, welche für das Versenden der Nachricht verantwortlich ist. Dies könnte beispielsweise die Adresse einer Sekretärin sein.

3. Der `Reply-To`-Header kann mit Einschränkungen ebenfalls zu den Absenderangaben gezählt werden. Er gibt an, an welche Adresse eventuell Antworten zu richten sind.

4. Der `Resent-From`-Header hat die gleiche Bedeutung wie das From-Feld im Falle einer Umleitung. Er bezeichnet die Adresse des Autors, welcher für die Umleitung verantwortlich ist.

5. Für den `Resent-Sender`-Header gilt entsprechendes.

Da keiner dieser Header vom empfangenden SMTP-Server hinzugefügt, sondern vom sendenden Server nach dem `DATA`-Befehl gesendet wird, können die Inhalte beliebig gefälscht werden.

Weitere Absenderinformationen können den sog. *Trace-Fields*, dem Return-Path und den Received-Headern entnommen werden. Der Return-Path enthält die Adresse aus dem letzten `MAIL FROM`-Befehl. Dieser Header wird vom letzten empfangenden Server hinzugefügt, der Inhalt kann aber durch entsprechend konfigurierte Sendesysteme beliebig gefälscht werden. An jeder Zwischenstation wird ein neuer Received-Header hinzugefügt; dieser enthält neben dem Hostnamen aus dem `EHLO`- bzw. `HELO`-Befehl und weiteren Details auch die IP-Adresse des sendenden Servers. Der Hostname kann wiederum durch spezielle Konfigurationen beliebig gefälscht sein. Die stärkste Aussagekraft hat die IP-Adresse[16], welche jedoch über Anonymisierungsdienste verschleiert werden könnte[17]. Da jedoch Received-Header nach dem `DATA`-Befehl übertragen werden, können speziell präparierte E-Mails mit beliebig gefälschten Received-Headern versehen werden, bevor die Nachricht in das Mailsystem eingeschleust wird. Dadurch wird dem tatsächlichen Übertragungsweg sozusagen ein Stück angehängt und damit dessen Überprüfung unmöglich gemacht. Letztendlich gibt es für den Empfänger keine praktikable Möglichkeit den Ursprung einer Nachricht zweifelsfrei festzustellen[18].

Fehlende End-to-End-Verbindung Wie im vorigen Abschnitt bereits angedeutet wurde, wird eine E-Mail fast immer über mehrere Zwischenstationen zum Empfänger gesendet. Als erstes erfolgt die Übertragung vom Host des Absenders zu seinem SMTP-Server. Dieser sendet die Nachricht an den dem Ziel am nächsten gelegenen SMTP-Server. Dies ist im Idealfall der des Empfängers, von dem dieser die Nachricht mittels eines POP- oder IMAP-Zugriffs empfangen kann. Um den Nachrichtentransfer gegen Störungen der Übertragungskanäle oder Serverausfälle abzusichern und so stabil wie möglich zu machen, können zwischen dem SMTP-Server des Senders und dem des Empfängers mehrere sog. SMTP-Relays liegen. Aber auch ohne diese Zwi-

[16] Laut RFC 2821 ist SMTP unabhängig vom zugrundeliegenden Protokoll. De facto wird aber bei weitem der größte Teil des Mailverkehrs über TCP/IP-Verbindungen übertragen.

[17] Z. B. JAP der TU Dresden [69].

[18] IP-Adressen sind nicht an bestimmte Hosts gebunden (Dial-Up-Verbindungen). Der Weg über die Protokolldateien eines ISP kann für den Empfänger nur als theoretisch bezeichnet werden.

schenstationen verhindert das Fehlen einer End-to-End-Verbindung, daß der Empfänger vom SMTP-Protokoll unabhängige Möglichkeiten nutzt, um sich über die Herkunft einer Nachricht Gewißheit zu verschaffen.

Kontrolle der Datenmenge Die Kontrolle der gesendeten und zu empfangenden Datenmenge unterliegt allein dem Absender. Der Empfänger hat nur die Möglichkeit, eingegangene Nachrichten zu löschen. Verstärkt wird dieses Problem durch das Expandieren der Empfängerlisten aus den To-, CC- und BCC-Feldern. Aus einer einzigen ausgehenden E-Mail können durch Angabe mehrerer Adressaten tausende oder sogar Millionen eingehende E-Mails werden. Dadurch werden der sowieso schon geringe Aufwand und die kaum vorhandenen Kosten für den Absender zum bedeutungslosen Faktor.

Protokollbasierte Rückmeldungen Im engen Zusammenhang mit obigem Abschnitt steht das Fehlen von protokollbasierten Rückmeldungen. Der Empfänger hat keine Möglichkeit, eine Nachricht zurückzuweisen oder anderweitig dem Absender bzw. dem sendenden MTA mitzuteilen, daß er die Übertragung abbrechen oder begrenzen soll.

2.4 Begriffsklärung

2.4.1 Spezielle SMTP-Anwendungen

Eine Anwendung welche SMTP implementiert kann je nach Einsatzzweck unterschiedliche Aufgaben erledigen, weswegen zur genaueren Unterscheidung verschiedene Begriffe gebräuchlich sind. Die im Rahmen dieser Arbeit verwendeten Begriffe werden anhand des folgenden Beispiels erläutert:

> Eine Mail wird von `pc12147` über `ma1.example.com` an `ma2.example.net` verschickt und von dort an `ma3.subd.example.net` weitergereicht, wo sie von einem User des Rechners `jmaxwell` abgeholt wird.

Server Als Server oder SMTP-Server wird stets der Partner einer SMTP-Sitzung bezeichnet, welcher auf eingehende Verbindungen wartet und

an welchen die Nachricht übertragen wird (`ma1.example.com`, `ma2.example.net`, `ma3.subd.example.net`).

Client Als Client oder SMTP-Client wird stets der Partner einer SMTP-Sitzung bezeichnet, welcher die Verbindung initiiert hat und welcher die Nachricht sendet (`pc12147`, `ma1.example.com`, `ma2.example.net`).

MTA Mail Transfer Agent (MTA) bezeichnet eine Software (bzw. Computer), welche über die gesamte SMTP-Funktionalität verfügt (`ma1.example.com`, `ma2.example.net`, `ma3.subd.example.net`).

MUA Mail User Agent (MUA) bezeichnet eine Software (bzw. Computer), welche nicht über die volle SMTP-Funktionalität verfügt. Typischerweise sind dies Anwendungen, mit welchen E-Mails verfaßt oder gelesen werden (`pc12147`, `jmaxwell`).

MDA Ein Mail Delivery Agent (MDA) ist in der Regel der letzte MTA auf dem Weg einer E-Mail, also jene Instanz, welche die Nachrichten in die Mailboxen der Empfänger verteilt bzw. von welcher ein MUA eine E-Mail abholt (`ma3.subd.example.net`).

MSA Ein Mail Submission Agent (MSA) ist in der Regel der erste MTA auf dem Weg einer E-Mail, also jene Instanz, an die ein MUA eine E-Mail weiterreicht (`ma1.example.com`).

Border-MTA Der letzte MTA bei Verlassen bzw. der erste MTA bei Erreichen einer Domain oder Organisation wird Border-MTA bezeichnet (`ma1.example.com`, `ma2.example.net`).

2.4.2 Weiterleitungsmechanismen

Der Begriff des Forwarding bzw. Weiterleiten wird in vielen unterschiedlichen Situationen verwendet, bei der eine Nachricht von einer Mailsoftware zu einer anderen geschickt wird. Dementsprechend viele Begriffe werden für diesen Vorgang verwendet, wie Umleiten, Weiterleiten, Relaying, Forwarding, Resending, Reintroducing, Redirecting und wohl noch einige mehr. Es ist nicht

Sinn dieses Abschnittes, mögliche Bedeutungen dieser Bezeichnungen zu erörtern, sondern die verschiedenen Fälle einer Weiterleitung darzustellen und im Einklang mit den einschlägigen RFCs klare Begriffsdefinitionen bereitzustellen.

Es gibt drei verschiedene Möglichkeiten, eine bereits empfangene Nachricht an ein anderes SMTP-System weiterzureichen, mit jeweils unterschiedlichen Auswirkungen auf die Headerzeilen.

1. Die E-Mail wird von einem Relay weitergereicht. Die Headerinformationen ändern sich bis auf das Hinzufügen eines Received-Headers nicht.

2. Weiterleitungen, welche der Benutzer durch spezielle Konfigurationen auf „seinem“ SMTP-Server erreicht (z. B. durch Einsatz von procmail, einer `.forward`-Datei für Sendmail, Mailinglisten). Je nach Methode und Software fallen die Änderungen an den Headerzeilen etwas unterschiedlich aus, in der Regel bleiben sie jedoch weitgehend unverändert; meist wird ein ‘Sender:’-Header hinzugefügt, manchmal auch weitere Resent-Header.

3. MUA-Software besitzt meistens eine Weiter- bzw. Umleitungsfunktionalität. Je nach Implementierung fallen die Änderungen an den Headerzeilen unterschiedlich aus.

Für diese drei Fälle einer Weiterleitung werden folgende Begriffsdefinitionen vorgeschlagen:

Relaying bezeichnet die Weiterreichung einer E-Mail durch einen MTA innerhalb des SMTP-Transportsystems auf dem Weg zum MDA des Empfängers. Das Relaying wird lediglich durch den Domain-Part der To-Adresse der eingehenden Mail, nicht aber durch den Local-Part beeinflußt.

Forwarding bezeichnet die Weiterleitung einer E-Mail an einen anderen SMTP-Server durch einen MDA oder eine Mailingliste. Welche Nachrichten einem Forwarding unterzogen werden, wird durch den Local-Part der To-Adresse der eingehenden Mail bestimmt.

Resending bezeichnet den Vorgang, wenn ein Benutzer unter Verwendung eines MUA die Kopie einer E-Mail an eine andere Adresse schickt.

Diese Definitionen decken sich weitgehend mit der Verwendung der Begriffe in den Abschnitten „3.4 Forwarding for Address Correction or Updating“, „3.7 Relaying“ und „3.10 Mailing Lists and Aliases“ des RFC 2821. Leider wurde bislang in keinem RFC spezifiziert, welche Headerzeilen bei welchem Weiterleitungsmechanismus wie zu ändern sind.

Im RFC 2822[19] findet sich folgende im Zusammenhang mit obigen Definitionen problematische Bemerkung, welche zur Klärung des Unterschieds zwischen Resending und Forwarding beitragen soll:

> „Reintroducing a message into the transport system and using resent fields is a different operation from “forwarding”. “Forwarding” has two meanings: One sense of forwarding is that a mail reading program can be told by a user to forward a copy of a message to another person, making the forwarded message the body of the new message. A forwarded message in this sense does not appear to have come from the original sender, but is an entirely new message from the forwarder of the message. On the other hand, forwarding is also used to mean when a mail transport program gets a message and forwards it on to a different destination for final delivery. Resent header fields are not intended for use with either type of forwarding.“

Die erste Erläuterung zur Bedeutung von Forwarding steht in gewisser Weise im Widerspruch zu der Aussage, daß Resent-Header jeder Nachricht hinzugefügt werden SOLLTEN, welche durch einen Benutzer erneut in das Transportsystem eingebracht wurde[20].

Die zweite Erläuterung entspricht eher dem Relaying, was an der Verwendung von „[...]for final delivery.“ festgemacht werden kann: die Nachricht ist also noch nicht an ihrem Bestimmungsort angekommen. Bei den auf Seite 32

[19] [82] S.27.

[20] „Resent fields SHOULD be added to any message that is reintroduced by a user into the transport system.“ [82] S.26.

beschriebenen Weiterleitungsfällen ist dies jedoch der entscheidende Unterschied zwischen dem ersten Fall (Relaying) einerseits und dem zweiten und dritten (Forwarding, Resending) andererseits. Bei einer Weiterleitung durch einen MDA bzw. MUA (Fälle zwei und drei) ist die E-Mail bereits an ihrem durch den ursprünglichen Absender vorgegebenen Ziel angekommen, was bei einem Relay nicht der Fall ist. Da obige Definitionen ansonsten im Einklang mit der Verwendung in anderen RFCs stehen, wird im Rahmen dieser Arbeit an diesen festgehalten und als mögliche Übersetzungen folgendes vorgeschlagen: relaying – weiterreichen, forwarding – weiterleiten, resending – wieder oder erneut (ver)senden bzw. umleiten.

2.5 Zusammenfassung

SMTP ist ein überaus leistungsfähiges Protokoll, welches sich für den Versand von Nachrichten bewährt hat: Es arbeitet stabil, zuverlässig und schnell. Daß Spam zu einem so großen Problem geworden ist, liegt daran, daß der Versand von Spam so einfach ist. Die Kosten und der Aufwand sind im Vergleich zu anderen Werbemaßnahmen (Postwurfsendungen, Plakate, etc.) zu vernachlässigen, mögliche Zieladressen finden sich leicht mittels entsprechend programmierter Agenten im WWW oder im Usenet, das Verschleiern oder Fälschen des Absenders – um möglicherweise drohenden gerichtlichen Konsequenzen zu entgehen – stellt ebenfalls keine Schwierigkeit dar. Und nicht zu vergessen: Spam hat Erfolg, d. h. durch E-Mail-Werbung läßt sich Geld verdienen. Wenn nur 0,001‰ aller in Deutschland erhaltenen Spam-Mails – also eine von 1 Million Nachrichten – zu einem Umsatz von 20€ führt, bedeutet dies bei einem täglichen Aufkommen von 1 Milliarde Werbemails eine durch Spam generierte Umsatzsteigerung von 20 000€. Dieses Rechenbeispiel dürfte der tatsächlichen Situation durchaus gerecht werden, wie verschiedene Meldungen und Urteile belegen[21].

Spam – im Sinne von unerwünschten Nachrichten – ist nicht vermeidbar. Die Initiative geht stets vom Absender (A) aus, deshalb ist der Empfänger (B)

[21] Siehe [39, 44, 103, 105].

gezwungen, Nachrichten zunächst anzunehmen, im Gegensatz beispielsweise zum HTTP-Protokoll, bei dem der Informationsfluß zwar ebenfalls von A nach B erfolgt, die Initiative aber vom Empfänger ausgeht, da die Information zuvor angefordert wurde.

Auch wenn es aufgrund dieses Prinzips immer zu unerwünschten Nachrichten kommen kann und wird, gibt es inzwischen vielversprechende Ansätze und Vorschläge, mit deren Hilfe verhindert werden kann, daß diese vereinzelten E-Mails zu einem Spamproblem werden.

You might be an anti-spam kook if you think your job is done after having explained the FUSSP (Final Ultimate Solution to the Spam Problem) to the IETF or The Industry.

aus einer Sammlung über die mit FUSSP verbundenen Schwierigkeiten [74]

3

Ansätze zur Spambekämpfung

In den letzten Jahren wurden zahlreiche Ideen entwickelt, welche einen Beitrag zur Eindämmung der Spamflut leisten sollen, auch wenn einige Vorschläge nicht explizit die Spambekämpfung zum Ziel haben. Im März 2003 wurde von der Internet Research Task Force (IRTF) die Arbeitsgruppe „Anti-Spam Research Group“ (ASRG) gegründet, mit dem Ziel, die Kräfte zu bündeln und die möglichen Maßnahmen gegen Spam voranzutreiben. Nach anfänglichem Enthusiasmus scheinen die Aktivitäten mittlerweile weitgehend im Sande zu verlaufen. Viele Links auf der Homepage der ASRG[22] sind verwaist, die letzten Neuigkeiten datieren aus dem Jahr 2004, und der Verkehr auf der ASRG-Mailingliste ist nahezu zum Erliegen gekommen. Allerdings haben sich mittlerweile auch andere Mailinglisten und Plattformen geöffnet, auf denen – zumeist jedoch nicht mehr unter der Regie der IRTF – weiterhin diskutiert wird. Problematisch ist dabei, daß viele eingereichte Drafts inzwischen nicht mehr auf den Seiten der IRTF oder ihrer Schwesterorgani-

[22] Siehe [3].

sation IETF (Internet Engineering Task Force) zu finden sind bzw. niemals im Rahmen einer Konferenz oder ähnlichem veröffentlicht wurden.

Zentraler Ansatz vieler Entwürfe ist es, die offensichtlich nicht ausreichenden Authentifizierungsmöglichkeiten des SMTP-Protokolls nachzubessern. Die hier aufgeführten Verfahren beschreiben Vorgehensweisen, welche es wahlweise ermöglichen sollen, den Absender oder den sendenden Mailserver oder wenigstens die Senderdomain zu authentifizieren.

Andere Vorschläge haben sich zum Ziel gesetzt, SMTP durch ein komplett neues Protokoll zu ersetzen, wie beispielsweise das Authenticated Mail Transfer Protocol (AMTP) von Bill Weinmann [101]. Solche Ansätze sind aus zweierlei Gründen fragwürdig. Zum einen könnte ein neues Protokoll vielleicht einige ursprüngliche Designziele von SMTP, welche sich nachträglich als problematisch herausgestellt haben, korrigieren. Aber wie schon auf Seite 34 erläutert wurde, können unerwünschte Nachrichten nicht gänzlich verhindert werden.

Zum anderen ist der Mailversand per SMTP der meistgenutzte Dienst im Internet mit vermutlich mehreren hundert Millionen Installationen. Ein neues Protokoll wird angesichts dieser breiten Basis und der engen Verzahnung mit Messaging-Diensten, Fax und SMS wenig Erfolg haben, sich durchzusetzen. Wie schwierig es ist, etablierte Systeme durch neue, deutlich leistungsfähigere zu ersetzen, zeigt nicht zuletzt die noch immer nicht erfolgte Umstellung von IPv4 auf IPv6.

3.1 Anforderungen an ein Anti-Spam-System

Eine Anti-Spam-System sollte folgenden Anforderungen genügen, um erfolgreich zu sein und weite Verbreitung zu erlangen:

1. Es muß unbekannten Absendern wenigstens den Versuch ermöglichen, einem beliebigen Empfänger einer beliebigen Domain eine E-Mail zu senden.

2. Der Transport der Nachricht ist nicht mit Kosten verbunden, auch keinen indirekten, wie beispielsweise Rechenzeit.

3. Es wird weder eine Zertifizierungsinstanz noch ein zentraler oder kommerzieller Verzeichnisdienst benötigt.

4. Nicht unmittelbar beteiligte Systeme dürfen nicht durch zusätzliche Aufgaben beeinträchtigt werden.

5. Der Versand oder Empfang von Nachrichten, welche keine für das Anti-Spam-System essentiellen Merkmale aufweisen, darf zumindest technisch nicht beeinflußt werden.

3.2 MTA-Autorisierung mittels DNS

Bemühungen zu diesem Thema wurden von der ASRG in der Arbeitsgruppe „MTA Authorization Records in DNS“ (MARID) gebündelt, welche im April 2004 gegründet wurde. Aufgrund patentrechtlicher Streitigkeiten im Zusammenhang mit dem von Microsoft eingereichten Vorschlag „Sender ID“ (siehe Seite 50) und des darin verwendeten Algorithmus zur Ermittlung des Absenders (siehe Seite 53) stellte die Arbeitsgruppe bereits im September des gleichen Jahres ihre Arbeit wieder ein.

Der grundlegende Gedanke der eingereichten Vorschläge ist, daß ein empfangender SMTP-Server anhand eines DNS-Lookups überprüfen können soll, ob der sendende MTA überhaupt berechtigt ist, Mails zu verschicken. Zur Speicherung der Informationen soll das DNS als verteilte Datenbank dienen. Das Ergebnis der Überprüfung soll dann möglicherweise bereits während der SMTP-Sitzung das Abweisen einer E-Mail ermöglichen.

3.2.1 „Repudiating Mail From“ und Nachfolger

Die Diskussion um eine Autorisierung des sendenden MTA mittels DNS wurde am 1. Juni 2002 durch ein Posting von David Green [26] auf der Namedroppers-Mailingliste angestoßen, worauf Paul Vixie mit seinem bereits im Mai verfaßten „Repudiating Mail From“ antwortete [100], welches nach eigener Aussage auf eine Idee von Jim Miller aus dem Jahr 1998 zurückgeht.

In den folgenden zwei Jahren wurde eine recht große Anzahl von Verfahren entwickelt, welche Vixies Idee aufgreifen: „Reverse MX Resource Record“ (RMX, Dez. 2002), „Designated Relays Inquiry Protocol“ (DRIP, Juni 2003), „Marking MTAs in Reverse DNS“ (MTAmark, Sep. 2003), „Designated Mailers Protocol“ (DMP, Dez. 2003) und „Flexible Sender Validation“ (FSV, Feb. 2004)[23]. Zu diesen Verfahren sind auch SPF und SenderID zu zählen, welche jedoch aufgrund ihrer größeren Bedeutung im Abschnitt 3.2.2 bzw. 3.2.3 separat vorgestellt werden.

3.2.1.1 Die Vorschläge im Überblick

Die hier besprochenen Verfahren machen sich zunutze, daß in der Regel die Zonendaten einer Domain nur von ihrem Inhaber geändert werden können. Dort abgelegte Informationen stammen also mit großer Sicherheit von einer autorisierten Person, sofern von Sicherheitslücken und Manipulationsmöglichkeiten im DNS abgesehen wird. Gemäß obiger Verfahren soll der Domaininhaber durch entsprechende Einträge im DNS veröffentlichen, welche MTAs autorisiert sind, Mails im Namen der Domain zu *versenden*, im Grunde also das umgekehrte Analogon zu einem MX-Resource Record, anhand dessen ermittelt wird, welcher MTA Mails für die Domain *empfangen* darf. Ein Mailserver kann also während einer SMTP-Sitzung durch eine DNS-Abfrage die Berechtigung des sendenden MTA überprüfen. Die hier besprochenen sechs Verfahren unterscheiden sich im Grunde lediglich in Aufbau und Typ der verwendeten Resource Records und darin, welche Informationen zur Überprüfung herangezogen werden; zur Verfügung stehen die Daten aus dem `MAIL FROM`-Befehl, der Hostname (inkl. Domain) aus dem `EHLO/HELO`-Befehl und die Client-IP-Adresse. Im folgenden werden alle sechs Verfahren mit ihren Resource Records und Eigenheiten kurz aufgeführt[24]:

Repudiating Mail From verwendet den Domainnamen aus dem `MAIL FROM`-Befehl. Der Fall, daß dieser leer ist – z. B. bei SMTP-Fehlermeldungen

[23] Siehe [8, 14, 21, 55, 93] .

[24] Die meisten Verfahren unterstützen auch explizit IPv6. Entsprechende RR wurden aus Gründen der Platzersparnis weggelassen.

–, wird nicht näher spezifiziert. Anhand des Domainnamens wird eine Liste der autorisierten Server bezogen. In weiteren DNS-Abfragen wird deren IP-Adresse ermittelt und mit der des Client verglichen.

```
$ORIGIN domain.tld.
MAIL-FROM MX 0 srvname1
          MX 0 srvname2
```

Codebeispiel 3.1: Resource Record-Beispiel: „Repudiating Mail From“

RMX verwendet den Domainnamen aus dem `MAIL FROM`-Befehl. Falls dieser leer ist, wird er aus den `EHLO/HELO`-Angaben ermittelt. RMX erlaubt oder verbietet den Versand von Mails anhand einzelner IP-Adressen, ganzer Netzbereiche (in CIDR-Notation) oder einzelner Hostnamen. Hierzu wird ein eigenes Tag-Value-Format verwendet, bei dem ein vorangestelltes ‘!’ das Sendeverbot kennzeichnet. Die IP-Adresse ist diejenige des sendenden Servers, der Hostname wird dem `EHLO/HELO`-Befehl entnommen. Darüber hinaus gibt es noch weitere Resource Records (z. B. Erlaubnis für alle MTAs, für die ein MX-RR existiert), welche die Konfiguration erleichtern, und es wird die Möglichkeit in Betracht gezogen, die Benutzung von Mailadressen durch DNS-Einträge zu autorisieren. RMX ist der einzige Vorschlag, welcher einen neuen RR-Typ einführt.

```
example.com  IN RMX ipv4:192.168.121.26
example.com  IN RMX ipv4:10.0.0.0/8
example.com  IN RMX !ipv4:1.2.3.4
example.com  IN RMX host:relay.provider.com
```

Codebeispiel 3.2: Resource Record-Beispiel: „RMX“ (Auswahl)

DRIP verwendet die Client-IP-Adresse und den vollständigen *Hostnamen* aus dem `EHLO/HELO`-Befehl.

```
*.IPv4.relays._email_.M.EXAMPLE.COM.           IN A 0.0.0.0
192_0_2_10.IPv4.relays._email_.M.EXAMPLE.COM.  IN A 192.0.2.10
```

Codebeispiel 3.3: Resource Record-Beispiel: „DRIP“

MTAmark verwendet lediglich die Client-IP-Adresse und kann auf weitere Informationen verzichten, da die Informationen im für das Reverse-Mapping zuständigen Ast (in-addr.arpa) des DNS und dort im neu geschaffenen Zweig

`_send._smtp._srv` abgelegt werden. In einem TXT-RR ist entweder eine 1 (Erlaubnis) oder eine 0 (Verbot) gespeichert. Zusätzlich wird vorgeschlagen, unter Verwendung des experimentellen RP-RR eine verantwortliche Person (responsible person) bzw. eine Mailadresse anzugeben ('@' wird zu '.').

```
$ORIGIN 0.0.10.IN-ADDR.ARPA.
1                       IN  PTR  mail.example.com.
_send._smtp._srv.1      IN  TXT  "1"
_smtp._srv.1            IN  RP   abuse.example.com.
```

Codebeispiel 3.4: Resource Record-Beispiel: „MTAmark"

DMP und FSV verwenden ebenfalls den Domainnamen aus dem `MAIL FROM`- bzw. `EHLO/HELO`-Befehl und die Client-IP-Adresse.

```
$ORIGIN example.com.
_smtp-client                    TXT "dmp="
*._smtp-client                  TXT "dmp=deny"
*.2.0.192.in-addr._smtp-client  TXT "dmp=allow"
```

Codebeispiel 3.5: Resource Record-Beispiel: „DMP"

FSV bietet darüber hinaus die Möglichkeit einzelne IP-Adressen und ganze Adreßblöcke in einem einzigen TXT-Record zu speichern oder jeweils einzelne A-Records anzulegen, was gemäß [55] den Vorteil haben soll, daß je nach Anwendung die Zahl der DNS-Abfragen reduziert wird.

```
$ORIGIN example.com
_fsv            TXT  "10.1.2.0/24" "10.9.9.9"
_fsv            A    0.0.0.2
*.2.1.10._fsv   A    127.0.0.2
```

Codebeispiel 3.6: Resource Record-Beispiel: „FSV"

Die Daten im Adreßteil der A-Records sind eine FSV-Besonderheit und keine IP-Adressen. Der erste A-Record steht im Zusammenhang mit dem TXT-Record. Einerseits soll damit zum Ausdruck gebracht werden, daß FSV-Informationen vorliegen, andererseits geben die beiden letzen Bytes der „IP-Adresse" an, wieviele Zeichenketten im TXT-Record gespeichert sind. Die Adresse 127.0.0.2 bringt die Sendeerlaubnis zum Ausdruck.

DRIP, DMP und FSV bieten die Möglichkeit mitzuteilen, daß die Domain prinzipiell das jeweilige Autorisierungsverfahren unterstützt, jedoch keine IP-Adresse eine Sendeerlaubnis hat[25].

[25] Bei DRIP und DMP ist dies jeweils der erste aufgeführte RR, bei FSV der zweite.

3.2.2 Sender Policy Framework

Dieses Verfahren hat eine recht turbulente Entwicklungsgeschichte, weswegen zunächst ein kurzer Überblick über den Entstehungsprozeß von SPF gegeben wird. Die ersten Fassungen des damals noch „Sender Permitted From“ genannten Verfahrens stammen aus dem Jahr 2003, im Mai 2004 wurde es dann unter dem Titel „Sender Policy Framework“ (SPF) von Mark Lentczner und Meng Weng Wong als Draft bei der Arbeitsgruppe MARID eingereicht. Infolge der Ähnlichkeit mit dem von Microsoft stammenden „Caller ID“, wurden beide Verfahren zusammengefaßt und als „Sender ID“ (siehe 3.2.3) veröffentlicht. Dieses ist auch als SPF 2.0 bekannt, während das ursprüngliche SPF auch „SPF Classic“ genannt wird. Wegen der bereits erwähnten Patentstreitigkeiten um Bestandteile von „Sender ID“ und der damit verbundenen Auflösung der MARID-Arbeitsgruppe trennten sich beide Verfahren wieder. Grundlage dieses Abschnitts ist die Fassung von Meng Weng Wong und Wayne Schlitt vom April 2006, welche als experimentelles RFC mit der Nummer 4408 bei der IETF erschienen ist[26].

Die grundlegende Funktionsweise ist die gleiche wie bei den in Abschnitt 3.2.1 vorgestellten Verfahren: Anhand eines DNS-Lookups wird überprüft, ob der SMTP-Client berechtigt ist, den Domainnamen zu verwenden, welcher im `MAIL FROM`-Befehl des Envelope angegeben wurde. Falls dieser keine Adresse enthält, wird analog zu oben der `EHLO/HELO`-Befehl ausgewertet. Allerdings ist SPF deutlich ausgereifter und bietet differenziertere Möglichkeiten zur Konfiguration der DNS-Einträge (z. B. Makrofunktionalität).

3.2.2.1 Die `check_host`-Funktion und SPF-Records

Zentrales Element von SPF ist eine (imaginäre) `check_host`-Funktion, welche für die DNS-Abfrage(n) und die Ergebnisermittlung zuständig ist. Diese benötigt als Eingangsparameter die IP-Adresse des sendenden MTA (`<ip>`), den gesamten Inhalt (`<sender>`) und den Domänenanteil (`<domain>`) des `MAIL FROM`- bzw. `EHLO/HELO`-Befehls. Von rekursiven Auswertungen abgesehen (siehe `include` bzw. `redirect`), stimmt in der Regel der Domänenanteil

[26] Siehe [104].

von `<sender>` mit `<domain>` überein. Die `check_host`-Funktion kann folgende Ergebnisse haben:

None Es kann nicht bestimmt werden, ob der sendende Client autorisiert ist oder nicht. Entweder konnte aus den Daten keine überprüfbare Senderdomain ermittelt werden, oder es wurden für diese keine DNS-Einträge veröffentlicht.

Neutral Der Besitzer der Domäne kann oder will keine Aussagen treffen, ob die betreffende IP-Adresse autorisiert ist. Dieses Ergebnis muß genauso wie „None“ verwendet werden. Dies ist z. B. für Tests zum Gebrauch von SPF-Records oder in der Einführungsphase wichtig.

Pass Der sendende Client ist autorisiert, den Domainnamen zu benutzen, die Domäne ist im Hinblick auf ihre Reputation für das Senden der Nachricht verantwortlich[27].

Fail Der sendende Client ist nicht autorisiert, den Domainnamen zu benutzen. Aufgrund dieses Ergebnisses kann die Mail besonders gekennzeichnet oder auch zurückgewiesen werden.

SoftFail ist als Zwischenstufe zwischen „Fail“ und „Neutral“ zu behandeln. Die Nachricht sollte nicht allein aufgrund dieses Ergebnisses zurückgewiesen, sondern einer genaueren Prüfung unterzogen werden.

TempError Bei der Prüfung ist ein vorübergehender Fehler aufgetreten (z. B. Server-Fehler oder Zeitüberschreitung bei DNS-Abfragen). Die Nachricht kann angenommen oder zeitweilig zurückgewiesen werden (SMTP-Antwort: “451 Requested action aborted: error in processing”).

PermError Die für die Domäne veröffentlichten Resource Records konnten nicht korrekt interpretiert werden (z. B. Syntaxfehler). Zur Behebung des Fehlers ist der Eingriff eines Benutzers (Administrators) erforderlich.

[27] Wörtlich: „The domain can now, in the sense of reputation, be considered responsible for sending the message.“ [104] S.8.

Das Ergebnis soll möglichst unter Verwendung des mittlerweile bei der IANA registrierten Received-SPF-Headers in den Kopfzeilen der Nachricht gespeichert werden. Darin *muß* das exakte Ergebnis („Pass“, „Fail“, „Neutral“ usw.) und *sollten* weitere Informationen wie `<ip>`, `<sender>` und `<domain>` festgehalten werden[28]. Leider ist die Angabe von Absenderinformationen nicht verpflichtend festgeschrieben. Damit das Ergebnis für den Empfänger nachvollzieh- und überprüfbar ist und für Reputationssysteme sinnvoll genutzt werden kann, wäre es wünschenswert, wenn die exakte Angabe möglichst vieler Absenderinformationen verpflichtend festgeschrieben wäre (MUST statt SHOULD). Darüber hinaus ist die aktuelle Syntax des Received-SPF-Headers schwierig zu parsen; auch hier wäre eine Vereinfachung wünschenswert.

Die `check_host`-Funktion überprüft zunächst den in `<domain>` angegebenen Domainnamen auf Korrektheit und führt dann eine entsprechende DNS-Abfrage durch. Die Antworten werden anhand eines zweistufigen Verfahrens gefiltert: Zunächst werden alle Ergebnisse entfernt, deren Versionsangabe nicht „v=spf1“ entspricht. Sollten unter den verbleibenden noch solche vom Typ SPF sein, werden in einem zweiten Schritt jene vom Typ TXT entfernt. Nach diesem Vorgang sollte genau ein Record übrigbleiben. Sind mehrere übrig, terminiert die Funktion mit dem Ergebnis „PermError“, bleibt keiner übrig, wird die Funktion mit „None“ beendet.

Die SPF-Informationen können wahlweise als TXT-RR oder in Form des neu definierten und bei der IANA registrierten SPF-RR veröffentlicht werden. Es wird empfohlen, beide gleichzeitig zu verwenden, welche dann exakt übereinstimmen müssen. Die Syntax des Inhalts ist für beide Typen identisch:

```
record    = version terms *SP
version   = "v=spf1"
```

Codebeispiel 3.7: ABNF für den Inhalt eines SPF-Records

Die Ausdrücke (engl. *terms*) bestehen aus „Mechanismen“ und „Modifikatoren“. Modifikatoren haben die Aufgabe, zusätzliche Informationen bereitzustellen, während anhand der Mechanismen festgelegt wird, welche Informationen und Algorithmen zu verwenden sind. Mechanismen kann ein „Qualifier“

[28] Zur genauen Syntax siehe [104] Kapitel 7.

vorangestellt sein, welcher das von `check_host` zurückzuliefernde Ergebnis angibt, falls ein Mechanismus erfolgreich ausgewertet werden kann. Ist kein Qualifier angegeben, wird als Standardwert '+' angenommen. Die möglichen Qualifier sind in Tabelle 3.1 aufgeführt.

Qualifier	Ergebnis
+	Pass
-	Fail
~	SoftFail
?	Neutral

Tab. 3.1: SPF-Qualifier für Mechanismen

Bevor die einzelnen Mechansimen erläutert werden, soll das bisher Gesagte anhand folgender Beispielkonfiguration veranschaulicht werden:

```
exple.com.           TXT "v=spf1 +mx a:colo.exple.com/28 -all"
smtp-out.exple.com.  TXT "v=spf1 a -all"
```

Codebeispiel 3.8: Resource Record-Beispiel: „SPF“

Der erste Record legt fest, daß in der Domäne `example.com` alle Mailserver mit existierendem MX-Resource Record und jene Hosts autorisiert sind, deren IP-Adresse in den durch `colo.exple.com/28` definierten Bereich fällt (Mechanismen `mx` und `a`), während alle anderen Hosts nicht autorisiert sind (Mechanismus `all`). Mit dem zweiten Record sind ausschließlich die Hosts zugelassen, welche einen A-Record für `smtp-out.exple.com` besitzen. Modifikatoren wurden in diesem Beispiel nicht verwendet. Die Auswertung der Mechansimen erfolgt der Reihe nach von rechts nach links. Sobald ein Mechanismus erfolgreich evaluiert werden kann, wird der Vorgang abgebrochen und der voranstehende Qualifier des Mechanismus bzw. der Standardwert als Ergebnis der `check_host`-Funktion zurückgegeben.

3.2.2.2 Mechanismen

Manche der im folgenden vorgestellten Mechanismen und Modifikatoren verwenden ein mit `<domain-spec>` bezeichnetes Konstrukt. Diese Zeichenkette kann Makrobefehle enthalten, deren Erläuterung an dieser Stelle jedoch zu weit führen würde. Wichtig ist dabei lediglich, daß durch Auswertung

eventuell vorhandener Makros ein vollqualifizierter Domainname (FQDN) entsteht, welcher mit `<target-name>` bezeichnet wird. Falls der Parameter `<domain-spec>` optional ist, wird als Standardwert der `<domain>`-Parameter der `check_host`-Funktion verwendet. Bei den Angaben zur Syntax bedeuten eckige Klammern ein optionales Argument, spitze Klammern bezeichnen einen Ausdruck, welcher im Anhang A „Collected ABNF“ des RFC 4408 definiert ist. Hiervon werden folgende verwendet: `<ip4-cidr-length>` bzw. `<ip6-cidr-length>` bezeichnen ein CIDR-Suffix für IPv4- bzw. IPv6-Adressen, `<dual-cidr-length>` bezeichnet allgemein ein CIDR-Suffix (IPv4 oder IPv6). `<ip4-network>` bezeichnet ein IPv4-Netzwerk in der üblichen DDN, `<ip6-network>` bezeichnet selbiges für IPv6 in den hierfür üblichen Schreibweisen.

all *Syntax:* `all`
Dieser Mechanismus wird immer erfolgreich evaluiert, d. h. er paßt zu jeder IP-Adresse. Da auf `all` folgende Mechansimen nicht ausgewertet werden, sollte er als letzter aufgeführt werden. Durch seine Verwendung kann der gesamte Auswertungsprozeß explizit beendet und ein Standardergebnis festgelegt werden.

include *Syntax:* `include:<domain-spec>`
Damit wird einen erneuter Aufruf der `check_host`-Funktion bewirkt, diesmal jedoch mit den Parametern `<ip>`, `<sender>` und `<target-name>`[29]. Mit diesem Mechanismus können administrative Grenzen überwunden werden. (Beispielsweise wenn eine Domäne `example.com` für abgehende Mails die Server der Domäne `example.net` verwendet.) Dieser Mechanismus gilt nur dann als erfolgreich ausgewertet, wenn `check_host` das Ergebnis „Pass“ liefert. Innerhalb eines Verwaltungsbereichs sollte die Verwendung von `include` minimiert werden; stattdessen ist der Modifikator `redirect` vorzuziehen.

[29] Die Autoren merken an, daß „include“ eine ungünstige Bezeichnung ist und daß „if-pass“ oder „on-pass“ eine bessere Wahl gewesen wäre.

a *Syntax:* `a [:<domain-spec>] [<dual-cidr-length>]`
Mit diesem Mechanismus wird eine DNS-Anfrage nach `<target-name>` gestellt und `<ip>` mit den zurückgelieferten Adressen bzw. dem Adreßbereich, verglichen.

mx *Syntax:* `mx [:<domain-spec>] [<dual-cidr-length>]`
Dieser Mechanismus kann erfolgreich ausgewertet werden, wenn `<ip>` ein MX-Host ist. Zuerst wird eine MX-Abfrage für `<target-name>` gestellt und anschließend für jeden zurückgelieferten Namen eine A-Abfrage. Jede Antwort-Adresse wird mit `<ip>` verglichen. Um Denial-of-Service-Angriffe zu vermeiden, dürfen nicht mehr als zehn MX-Namen aufgelöst werden.

ptr *Syntax:* `ptr [:<domain-spec>]`
Dieser Mechanismus überprüft, ob für `<ip>` ein Reverse-Mapping existiert und auf die korrekte Domäne verweist. Hierzu wird zunächst ein Reverse-Lookup für `<ip>` und anschließend ein PTR-Lookup für jeden zurückgelieferten Namen durchgeführt, um diesen zu überprüfen. Zur Vermeidung von Denial-of-Service-Angriffen dürfen wiederum nicht mehr als zehn Namen überprüft werden. Der Mechanismus ist erfolgreich, wenn die überprüften Namen die Endung `<target-name>` haben. Da der Vorgang sehr aufwendig und langsam ist, sollte er nur selten eingesetzt werden und möglichst weit rechts in der Liste der Mechanismen stehen.

ip4 und **ip6** *Syntax:* `ip4:<ip4-network> [<ip4-cidr-length>]`
bzw. `ip6:<ip6-network> [<ip6-cidr-length>]`
Dieser Mechanismus wird erfolgreich ausgewertet, wenn `<ip>` innerhalb des angegebenen Netzwerks liegt. Falls kein CIDR-Suffix angegeben ist, wird automatisch /32 bzw. /128 verwendet.

exists *Syntax:* `exists:<domain-spec>`
Hier wird der aus `<domain-spec>` resultierende Domainname für einen A-Lookup verwendet. Wenn irgendein A-Record in der Antwort enthalten ist, gilt die Auswertung als erfolgreich. Dieser Mechanismus ermöglicht beliebig

komplexe Abfragen unter Verwendung frei wählbarer Bestandteile des Envelopes. Beispielsweise könnte `<domain-spec>` aus

```
v=spf1 exists:%{ir}.%{l1r+-}._spf.%{d} -all
```

zu `1.2.0.192.someuser._spf.example.com` ausgewertet werden. (`%{ir}` ist die umgekehrte IP-Adresse, `%{l1r+-}` ist ein Bestandteil des Local-Part von `<sender>`, und `%{d}` entspricht `<domain>`). Gibt es dazu einen A-Record, wäre die entsprechende IP-Adresse sendeberechtigt. Mit diesem Mechanismus sind also sehr genaue Konfigurationen und Abfragen realisierbar, welche im Prinzip sogar die Autorisierung einzelner User ermöglichen.

3.2.2.3 Modifikatoren

Modifikatoren sind Name/Wert-Paare, welche durch ein Gleichheitszeichen getrennt sind. Sie dürfen an beliebiger Stelle auftreten, sollten jedoch am Ende des SPF-Records stehen und dürfen jeweils nur einmal vorkommen. Ihre Reihenfolge ist im Gegensatz zu den Mechansimen nicht von Bedeutung.

redirect *Syntax:* `redirect=<domain-spec>`
Falls kein Mechanismus erfolgreich ausgewertet werden konnte, wird hiermit ähnlich wie durch den `<include>`-Mechanismus ein weiterer Aufruf der `check_host`-Funktion, allerdings mit `<target-name>` statt `<domain>`, bewirkt. Das Ergebnis dieses Aufrufs ist das Ergebnis des aktuellen Auswertungsvorgangs, mit der Ausnahme, daß statt „None“ „PermError“ zurückgegeben wird, falls kein SPF-Record gefunden wurde oder `<target-name>` keine korrekte Syntax hat. Mit diesem Modifikator können Organisationen die selben SPF-Records für verschiedene Domänen verwenden:

```
la.example.com.    TXT "v=spf1 redirect=_spf.example.com"
ny.example.com.    TXT "v=spf1 redirect=_spf.example.com"
sf.example.com.    TXT "v=spf1 redirect=_spf.example.com"
_spf.example.com.  TXT "v=spf1 mx:example.com -all"
```

Codebeispiel 3.9: Beispiel für die Verwendung des `redirect`-Modifikators

explanation *Syntax:* `exp=<domain-spec>`
Falls die `check_host`-Funktion das Ergebnis „Fail“ liefert, kann dieser Mo-

difikator verwendet werden, um ausführlichere Erklärungen zum Grund des Scheiterns mitzuteilen. Hierzu werden in `<domain-spec>` vorhandene Makros expandiert und ein TXT-Lookup für den resultierenden Domainnamen durchgeführt. Die zurückgelieferten Zeichenketten werden aneinandergehängt und eventuell darin enthaltene Makros ausgeführt. Der resultierende Text enthält die fertige Erläuterung.

3.2.3 Caller ID und Sender ID

Das im Februar 2004 bei der Arbeitsgruppe MARID eingereichte Verfahren „Caller ID“ [62] war das erste Autorisierungsverfahren, welches Informationen über den Absender nicht dem Envelope, sondern den Headerzeilen der E-Mail entnimmt. Der hierfür verwendete Mechanismus heißt „Purported Responsible Address“ und wird im Abschnitt 3.2.3.2 erläutert. Der zweite wesentliche Unterschied zu den bisher vorgestellten Verfahren ist die Speicherung der Autorisierungsinformationen. Hierfür ist ein XML-Dokument vorgesehen, welches – eventuell aufgeteilt in mehrere Fragmente – in Form von TXT-Records im DNS veröffentlicht wird. Verwunderlich ist dabei, daß ausgerechnet das platzraubende XML-Format gewählt wurde, während der zur Verfügung stehende Speicherplatz in TXT-Records eher gering ist, da die meisten Lookups und Antworten per UDP transportiert werden und nicht mehr als 512 Bytes Nutzdaten enthalten dürfen[30]; zusätzlich muß auf bereits existierende TXT-Records Rücksicht genommen werden. Offensichtlich ist diese Idee bald wieder verworfen worden, denn in der ersten Fassung von Sender ID, welches ja durch Zusammenführen von Caller ID und SPF entstand, ist der Ansatz, XML-Dokumente zu verwenden, wieder verschwunden.

3.2.3.1 Unterschiede zwischen Sender ID und SPF

Ebenso wie SPF ist Sender ID mittlerweile als experimentelles RFC unter der Nummer 4406 erschienen [57]. Sender ID stellt eine Erweiterung von SPF dar, welche es ermöglicht, statt der `MAIL FROM`-Adresse die aus den Headerzeilen ermittelte Absenderadresse (PRA, siehe 3.2.3.2) für die Autorisierung

[30] [64] S.32.

zu verwenden.

Die Veröffentlichung erfolgt ebenfalls als TXT- oder SPF-Record im DNS. Der Inhalt eines entsprechenden Resource Records wird wie folgt definiert:

```
record      = version terms *SP
version     = "v=spf1" | ( "spf2." ver-minor scope)
ver-minor   = 1*DIGIT
scope       = "/" scope-id *( "," scope-id )
scope-id    = "mfrom" / "pra" / name
```

Codebeispiel 3.10: ABNF für den Inhalt eines Sender ID-Records

Anhand dieser Definitionen wird zweierlei ersichtlich. Zum einen ist erkennbar daß die volle SPF-Funktionalität erhalten bleibt bzw. erhalten bleiben muß, nämlich wenn als Versionsangabe „v=spf1“ verwendet wird. Zum anderen wird an dieser Stelle die Sender ID-Funktionalität „angeflanscht“, welche über die Versionsangabe „spf2.“ eingeleitet wird und mit deren Bereichsangabe (scope) festgelegt wird, für welche Art der Autorisierung der Datensatz verwendet werden kann. Wann welche Autorisierungsart zu verwenden ist, wird durch das RFC 4406 nicht näher festgelegt; es ist also nur von der Software abhängig, ob ein `MAIL FROM`-Test (de facto also Autorisierung gemäß SPF) oder ein PRA-Test durchzuführen ist[31]. Die möglichen Ausdrücke (terms) bleiben unverändert, lediglich die Funktionsweise der Modifikatoren wird etwas verändert. Sender ID unterscheidet zwischen global wirksamen Modifikatoren und solchen, deren Wirkung positionsabhängig ist, sowie zwischen solchen, welche mehrfach oder nur einmal (singulär) auftreten dürfen. Neue Modifikatoren werden nicht definiert und die bereits vorhandenen (`exp` und `redirect`) als singulär und global festgelegt.

Die von SPF stammende `check_host`-Funktion erhält einen zusätzlichen Parameter `<scope>`, anhand dessen ihr Verhalten spezifiziert wird. Je nach dessen Wert wird `<sender>` bzw. `<domain>` aus dem `MAIL FROM`-Befehl ent-

[31] Diese Vorgehensweise erinnert an die Microsoft oftmals vorgeworfene Geschäftspraxis, Konkurrenten dadurch zu verdrängen, daß deren Konzepte und Verfahren zunächst übernommen und später verändert werden, wobei die Grenzen und Unterschiede zunehmend verblassen. Eine deutlichere Abgrenzung wäre beispielsweise durch

```
record = version terms *SP
version = "sendID" ver "." ver "/" scope *( "," scope )
ver = 1*DIGIT
scope = "mfrom" / "pra" / name
```

möglich gewesen.

nommen oder gemäß PRA ermittelt. Ansonsten unterscheiden sich die beiden `check_host`-Funktionen nur in ihrer Reaktion auf den DNS-Fehler „domain does not exist“ (RCODE 3) (bei SPF wird „None“ geliefert, bei Sender ID für einen PRA-Test „Fail“) und der Auswahl des Resource Records, falls bei der ersten DNS-Anfrage mehrere Ergebnisse geliefert wurden. Im Gegensatz zu dem zweistufigen Verfahren bei SPF werden die Antworten in fünf Schritten gefiltert:

1. Falls unter den Antworten Records vom Typ SPF sind, werden alle TXT-Records entfernt.

2. Alle Records, welche mit ungültigen Versions- und Bereichsangaben beginnen, werden verworfen.

3. Von den Records mit der Versionsangabe „spf2“ werden jene entfernt, bei denen keine Bereichsangabe mit der des `<scope>`-Parameters der `check_host`-Funktion übereinstimmt.

4. Falls die DNS-Anfrage zwei Records liefert, einen mit Version „v=spf1“ und einen mit „spf2“, wird jener mit der Angabe „spf2“ verwendet. Falls dessen Bereichsangabe nicht zu der von `<scope>` passt, wird der Record mit der Version „spf1“ ausgewählt.

5. Falls kein „spf2“-Record die passende Bereichsangabe besitzt und kein „v=spf1“-Record vorhanden ist, wird kein Record ausgewählt.

Die Schritte vier und fünf sind mißverständlich, da bereits im dritten Schritt „spf2“-Records ohne passende Bereichsangabe entfernt wurden, aber genau dieser Fall angesprochen wird, d. h. in Schritt vier und fünf können keine „spf2“-Records übrig sein, welche keine passende Bereichsangabe haben. Die einzig sinnvolle Interpretation ist, daß ab Schritt vier von einem anderen Fall gesprochen wird und gleich zu Beginn eine Unterscheidung vorzunehmen ist: „Gibt es genau zwei Records, dann fahre mit Schritt 4. fort, ansonsten mit Schritt 1.“ Leider gibt es keinen Unterschied zu früheren Fassungen, so daß in diesem Punkt keine Klarheit zu erzielen ist.

Ebenso ist nicht recht nachvollziehbar, warum eine Schwäche von Sender ID nicht behoben wurde, obwohl sie im RFC im Abschnitt „6.5. Malicious DNS Attacks on Third Parties“ beschrieben wird: Ein Angreifer ersetzt die PRA-Angaben einer Nachricht mit denen seines Opfers und schickt die E-Mail an einen unbeteiligten Mailempfänger, wo der Sender ID-Test wie beabsichtigt fehlschlägt. Wird die Nachricht während der SMTP-Session überprüft, kann diese abgebrochen werden. Wird jedoch die Nachricht zunächst angenommen und erst danach überprüft, ist gemäß Abschnitt „5.3. Fail“ eine Delivery Status Notification (DSN, auch Bounce genannt) zu verschicken. Darin kann die ursprüngliche Nachricht enthalten sein. Die Frage ist, an wen diese DSN zu verschicken ist. Der ursprüngliche Adressat kann daran kein Interesse haben; solche Nachrichten könnten als Spam aufgefaßt werden. Die PRA-Adresse kann aber ebenfalls nicht verwendet werden, da deren Überprüfung ja gerade fehlgeschlagen ist und somit nicht für den Nachrichtenversand verantwortlich gemacht werden kann. Eine an diese Adresse verschickte DSN wirkt dort vermutlich ebenfalls störend. Die einzig sinnvolle Lösung kann nur lauten, im Falle des Ergebnisses „Fail“ für eine bereits angenommene Nachricht diese nicht weiterzuleiten und keine sonstigen sich darauf beziehenden Nachrichten zu verschicken.

3.2.3.2 Purported Responsible Address

Purported Responsible Address (PRA) ist ein Verfahren, den mutmaßlichen Absender einer E-Mail aus den Headerzeilen zu ermitteln, und ist als RFC veröffentlicht worden [56]. In diesem Zusammenhang bedeutet „mutmaßlich“, daß Headerzeilen natürlich gefälscht werden können und deswegen keine Gewißheit bestehen kann, ob die ermittelte mit der tatsächlichen Absenderadresse übereinstimmt.

Zugleich stellt PRA den Grund für das Scheitern der Arbeitsgruppe MARID und für die Ablehnung von Sender ID innerhalb der ASRG dar, weil es Bestandteil eines Patentantrags[32] von Microsoft ist. Dabei entspricht dessen Vorgehensweise dem naheliegendsten und im Grunde einzig gangbaren Weg,

[32] Siehe [5].

wie aus den Headerzeilen der vermutliche Absender zu ermitteln ist:

1. Suche das erste nicht-leere Resent-Sender-Feld. Falls keines vorhanden ist oder falls diesem ein nicht leeres Resent-From-Feld, gefolgt von einem oder mehreren Received- oder einem Return-Path-Feld, vorausgeht, fahre mit Schritt 2 fort, ansonsten mit Schritt 5.

2. Suche das erste nicht-leere Resent-From-Feld. Falls keines vorhanden ist fahre mit Schritt 3 fort, andernfalls mit Schritt 5.

3. Suche alle nicht-leeren Sender-Header. Falls es keine gibt, mache mit Schritt 4 weiter. Gibt es genau einen, fahre mit Schritt 5 fort, gibt es mehrere, mit Schritt 6.

4. Suche alle nicht-leeren From-Header. Falls es genau einen gibt, fahre mit Schritt 5 fort, ansonsten mit Schritt 6.

5. Falls die Kopfzeile mißgebildet ist (z. B. mehrere Adressen, kein Domainname in der Adresse etc.) fahre mit Schritt 6 fort. Andernfalls ist die enthaltene Adresse die des vermutlichen Absenders (PRA).

6. Die Nachricht ist mißgebildet; es ist nicht möglich einen vermutlichen Absender (PRA) zu bestimmen.

3.2.4 Diskussion

Die Bekanntgabe, welche Rechner berechtigt sind, für eine Domain Nachrichten zu verschicken, ist ein erster sinnvoller Ansatz, um die mißbräuchliche Verwendung von Adressen einzuschränken. Da es sich um domainbezogene Informationen handelt, ist es naheliegend und sinnvoll, das DNS zur Veröffentlichung zu verwenden und hierfür das Analogon zu einem MX-RR zu schaffen.

Alle hier vorgestellten Verfahren haben mehr oder weniger stark mit den verschiedenen Weiterleitungsmöglichkeiten zu kämpfen. Systeme, welche auf `MAIL FROM`-Angaben zurückgreifen, erweisen sich vor allem beim Relaying als problematisch, weil die IP-Adresse des Relays möglicherweise nicht zu der

Domain der Absenderadresse gehört. Sender ID reagiert beim PRA-Test hingegen empfindlich auf ungewöhnliche, aber nicht böswillige Veränderungen der verschiedenen Absenderangaben, wie sie beim Forwarding und Resending vorgenommen werden. Wünschenswert wären verbindliche Regelungen, wie bei verschiedenen Weiterleitungsmöglichkeiten die Envelope- und Headereinträge anzupassen sind. Möglicherweise kann diese Problematik durch Einsatz von Verfahren wie das „Sender Rewriting Scheme“ (SRS, [88]) gelindert werden.

Sehr zu begrüßen ist die Verwendung von Received-SPF-Headerzeilen, weil dadurch das Ergebnis durch den Empfänger überprüfbar wird. Dabei ist jedoch darauf zu achten, daß nicht nur das Vorhandensein eines Received-SPF-Headers überprüft wird, sondern auch von wem dieser hinzugefügt wurde. Verläßlich sind nur jene, welche vom zustellenden MDA stammen.

Alles in allem stellen SPF und Sender ID zwei sinnvolle Möglichkeiten dar, die Autorisierung eines SMTP-Clients zu überprüfen, wobei Sender ID durch die Hinzunahme des PRA technisch gesehen das bessere, weil flexiblere Verfahren ist. Allerdings stehen mögliche zukünftige Patentansprüche Microsofts im Widerspruch zu einer Anerkennung als offenem und freiem Standard. Da im Bereich des Mailtransports sehr häufig Open-Source-Programme eingesetzt werden (z. B. Sendmail), wird Sender ID aufgrund der Lizenzproblematik von dieser Seite möglicherweise keine Unterstützung erfahren, was wiederum einer schnellen und weiten Verbreitung abträglich ist.

3.3 Signaturverfahren

Die in diesem Abschnitt vorgestellten Verfahren beruhen auf der Berechnung einer Signatur anhand eines Public-Key-Algorithmus und einer Einweg-Hashfunktion[33]. Hierbei wird für das zu signierende Dokument ein Hashwert errechnet und dieser mittels des privaten Schlüssels chiffriert. Zur Überprüfung des erhaltenen Dokuments wird dessen Hashwert berechnet, die Signatur anhand des öffentlichen Schlüssels dechiffriert und das Ergebnis mit dem er-

[33] Für eine ausführlichere Erläuterung der genauen Funktionsweise von Public-Key-Chiffren und Einweg-Hashfunktionen siehe [87] S.491ff und 525ff.

rechneten Wert verglichen. Sind beide identisch, ist dadurch gesichert, daß die Nachricht während des Transports nicht verändert wurde und – sofern die Absenderangaben mit dem Besitzer des öffentlichen Schlüssels übereinstimmen – tatsächlich vom angegebenen Absender stammt. Diese Verfahren dienen also vornehmlich der Autorisierung und sollen Adreßmißbrauch verhindern. Der Unterschied der hier vorgestellten Systeme liegt im wesentlichen in der Schlüsselverwaltung und der Art, wie die Signatur der E-Mail hinzugefügt wird.

3.3.1 Pretty Good Privacy und Verwandte

Das erste Verfahren dieser Art beruht auf der Software Pretty Good Privacy (PGP) von Phil Zimmermann aus dem Jahr 1991 und wurde 1996 als RFC 1991[34] veröffentlicht. Ungefähr zu gleichen Zeit entstanden die nach dem gleichen Prinzip funktionierenden Protokolle S/MIME[35] und OpenPGP[36].

Allen drei Verfahren ist gemeinsam, daß sie de facto eine zentrale Instanz zur Verwaltung der Schlüssel (bzw. Zertifikate bei S/MIME) benötigen. Um eine Nachricht zu überprüfen, benötigt der Empfänger den öffentlichen Schlüssel, welcher am einfachsten über zentrale Keyserver bezogen werden kann. Des weiteren ist ein mit einem Namen oder einer Adresse verknüpftes Schlüsselpaar nur dann aussagekräftig, wenn die Übereinstimmung von Name und Besitzer überprüfbar ist (Zertifizierungsinstanzen, „Web of Trust“).

Obwohl diese Technik im Kern kein Verfahren zu Spambekämpfung ist, wird sie doch immer wieder mit diesem Problem in Verbindung gebracht. Voraussetzung hierfür wäre jedoch, daß ein sehr großer Anteil des Mailverkehrs aus signierten (und verschlüsselten) Nachrichten besteht. Dies bedeutet wiederum, daß nahezu für jede Mailadresse ein Schlüsselpaar existiert. Unter diesen Voraussetzungen wäre es möglich, den Absender zu identifizieren, die Authentizität der E-Mail und die Autorisierung zu überprüfen. Daran, daß wichtige und sicherheitsrelevante E- Mails (z. B. Microsoft Security Bulletin) häufig durch diese Technik geschützt sind, wird sichtbar, daß das Verfahren

[34] Siehe [4].
[35] Siehe [80, 81].
[36] Siehe [10].

prinzipiell funktioniert. Es ergeben sich jedoch folgende Probleme:

1. Der Aufwand für die öffentlichen Keyserver wäre beträchtlich (aber handhabbar).

2. Die Technik erfordert einen sensiblen Umgang mit den Schlüsseln und damit kryptographische Kenntnisse seitens der Benutzer. Bei mehreren hundert Millionen notwendigen Schlüsselpaaren führt dies zu einem großen Anteil kompromittierter Schlüssel, wodurch das Verfahren seine Stärke verliert (Lücken im „Web of Trust“).

3. Eine Authentifizierung des Autors ist nicht nötig, eine Autorisierung des Absenders ist ausreichend. DKIM erreicht mit deutlich weniger Aufwand fast das gleiche Ziel.

4. In vielen Ländern ist der Einsatz starker kryptographischer Software verboten.

Auch wenn der Gedanke im ersten Augenblick bestechen mag, wird ersichtlich, daß sich das Spamproblem anhand dieser Methode nicht lösen läßt. Der beste Beweis ist, daß die Technik seit vielen Jahren einsatzbereit ist, sich aber nicht durchsetzen konnte. Da Versender von Spam nichts unversucht lassen, um trotzdem ihre Nachrichten ans Ziel zu bringen, wäre der Schaden für den eigentlichen Einsatzbereich dieser Software möglicherweise größer als der Nutzen bei der Bekämpfung von Spam.

Etwas weniger weit greifen die beiden Verfahren „DomainKeys“ und „Identified Internet Mail“ (DK bzw. IIM) welche in den letzten Jahren unabhängig voneinander entstanden sind. Aufgrund ihrer starken Ähnlichkeit wurden sie 2005 zu „DomainKeys Identified Mail“ (DKIM) zusammengefaßt. Der grundlegende Gedanke hierbei ist, daß nicht unbedingt für jede Adresse ein eigenes Schlüsselpaar existieren muß, sondern daß es ausreichend ist, für einzelne Gruppen oder Arbeitsbereiche einer Domain bzw. Organisation einen gemeinsamen Schlüssel zu verwenden, anhand dessen überprüft werden kann, daß die Nachricht tatsächlich von dieser Gruppe, Domain, Organisation etc. stammt. Eine Nachricht muß hierbei nicht durch den Verfasser einer E-Mail

unterzeichnet werden. Vielmehr ist vorgesehen, daß jede ausgehende Nachricht spätestens durch den Border-MTA der Domain signiert wird.

3.3.2 Grundlegende Funktionsweise von DK, IIM und DKIM

Alle drei Verfahren verfolgen das Ziel, Adreßmißbrauch zu verhindern, indem die sendende Domain authentifiziert werden kann. Hierzu wird vor dem Versand einer E-Mail aus ausgewählten Headerzeilen und dem Body ein Hashwert errechnet und dieser mittels eines privaten RSA-Schlüssels signiert[37]. Der resultierende Wert wird zusammen mit einigen Verfahrensparametern in einer eigenen Headerzeile der E-Mail vorangestellt. Der öffentliche Teil des RSA-Schlüssels wird per DNS oder durch einen domain-eigenen Keyserver publiziert, der private Schlüssel steht nur der unterzeichnenden Instanz zur Verfügung. Diese kann sowohl lokal auf dem Computer eines Benutzers als auch auf dem zentralen Mailausgangsserver installiert sein. Im zweiten Fall ist es jedoch zwingend erforderlich, dass sich der Benutzer gegenüber dem MTA authentifiziert hat[38].

Zur Überprüfung einer E-Mail wird die Headerzeile gesucht, welche die Signatur enthält. Darin enthalten sind alle nötigen Parameter, um mit Hilfe eines DNS-Lookups oder einer Anfrage an den Keyserver den öffentlichen Schlüssel der Domain zu erhalten. Mit diesem können die Signatur und somit der Absender bzw. die sendende Domain überprüft werden. Im Normalfall sollte nur der Domaininhaber (bzw. eine von ihm autorisierte Person) in der Lage sein, die notwendigen Daten im DNS oder auf dem Keyserver zu veröffentlichen. Da sich der Absender gegenüber der Domain authentifiziert haben muß, ist gewährleistet, daß die Absenderadresse mit dem tatsächlichen Absender übereinstimmt.

Zur Berechnung des Hashwertes wird die E-Mail in eine kanonische Form[39]

[37] In allen Vorschlägen wird eine Kombination aus SHA1 (bzw. SHA256) und RSA verwendet; andere kryptographische Verfahren mit dem gleichen Leistungsspektrum sind aber in späteren Versionen denkbar, wie in [1] erwähnt wird.

[38] Z. B. Authentifizierung nach RFC 2554 [66] oder POP vor SMTP, etc.

[39] Auch Normalform genannt.

gebracht. Dieser Schritt soll gewährleisten, daß die Signatur einer Nachricht gültig bleibt, auch wenn sie während des Transports leichten Veränderungen unterworfen wird, wie beispielsweise geänderte Zeilenumbrüche oder eine geänderte Reihenfolge der Headerzeilen. Die eigentliche E-Mail bleibt bei diesem Vorgang unverändert, die kanonisierte Form wird nur zur Berechnung des Hashwertes benötigt.

Da es unterschiedliche Ansichten darüber gibt, welche Veränderungen an einer E-Mail während des Transports zulässig sein sollen, gibt es für jedes der drei Verfahren zwei unterschiedliche Vorgehensweisen. Jeweils eines ist toleranter gegenüber Veränderungen (`nofws` bzw. `relaxed`), während das andere (`simple` bzw. `plain`) keine oder nur sehr geringe Änderungen zuläßt. Folgende Aufstellung gibt einen Überblick für alle drei Verfahren:

DomainKeys Hier wird die gesamte E-Mail (Header und Body) kanonisiert.

simple Alle Zeilen werden in der Reihenfolge ihres Auftretens verarbeitet. Wenn das h-Tag (siehe Seite 62) verwendet wird, fließen nur diejenigen Headerzeilen in die Berechnung ein, die dort aufgeführt sind. Der Body wird vom h-Tag nicht beeinflußt. Alle Zeilenendezeichen werden durch ein CRLF ersetzt. Leere Zeilen[40] am Ende der Nachricht werden ignoriert. Dies gilt auch für die Leerzeile zwischen Header und Body.

nofws Alle Zeilen werden in der Reihenfolge ihres Auftretens verarbeitet. Umbrüche in Headerzeilen werden entfernt, so daß die Headerinformationen in jeweils einer einzigen Zeile stehen. Wenn das h-Tag verwendet wird, fließen nur die Headerzeilen in die Berechnung ein, die dort aufgeführt sind. Der Body wird vom h-Tag nicht beeinflußt. In jeder Zeile wird jedes auftretende „Folding Whitespace“ (FWS) durch je ein CRLF ersetzt. Ein FWS dient dazu, eine Headerzeile auf mehrere Zeilen zu verteilen. Es kann ein Whitespace oder ein CRLF sein, muß dann jedoch von einem Whitespace gefolgt werden. Leere Zeilen am Ende der Nachricht werden ignoriert.

[40] Zeilen, welche nach Entfernen des Zeilenendezeichens die Länge Null haben.

IIM Hier wird nur der Body kanonisiert, die Headerzeilen werden nicht berücksichtigt.

plain Alle Bytes des Body fließen ohne Modifikation in die Berechnung des Hashwertes ein.

nofws Alle Whitespaces werden entfernt und jeweils das achte Bit eines Bytes wird gelöscht.

DKIM Hier können die Kanonisierungsalgorithmen für Header und Body getrennt angegeben werden.

simple (Header) Die Headerzeilen werden in keiner Weise verändert, insbesondere werden keine Buchstaben in Kleinbuchstaben umgewandelt.

relaxed (Header) Die Namen der Headerfelder werden in Kleinbuchstaben umgewandelt. Umbrüche in den Headerzeilen werden entfernt, so daß die Headerinformationen in jeweils einer einzigen Zeile stehen (abschließende CRLF bleiben erhalten). Alle Folgen von einem oder mehreren Whitespaces werden durch ein einziges Leerzeichen ersetzt. Alle Whitespaces am Ende der Headerzeile und alle Whitespaces vor und nach dem Doppelpunkt, welcher den Headernamen von seinem Inhalt trennt, werden entfernt.

simple (Body) Außer der Entfernung von leeren Zeilen am Ende der Nachricht werden keine Veränderungen vorgenommen.

relaxed (Body) Alle Whitespaces am Ende einer Zeile werden ignoriert und alle aufeinanderfolgenden Whitespaces werden zu einem Leerzeichen zusammengefaßt. Alle leeren Zeilen am Ende der Nachricht werden entfernt[41].

Die Details, welche Informationen in welcher Form in die neu geschaffenen Headerfelder aufgenommen werden etc., werden in den nächsten Abschnitten für jedes Verfahren getrennt erläutert.

[41] Die Autoren sind sich nicht sicher, ob dieses Verfahren benötigt wird, ihrer Meinung nach müßte `simple` ausreichen. In späteren Versionen könnte dieser Abschnitt also fehlen.

3.3.3 DomainKeys (DK)

Die jüngste Fassung datiert aus dem März 2005 und wurde von Mark Delany, Mitarbeiter bei Yahoo!, als Internet-Draft[42] herausgegeben.

3.3.3.1 Bereitstellung des öffentlichen Schlüssels

Das DomainKeys-Verfahren verwendet TXT-Records, um den öffentlichen Schlüssel per DNS zu publizieren, ein Keyserver ist nicht vorgesehen. Ein entsprechender TXT-RR könnte beispielsweise folgende Gestalt haben:

```
brisbane._domainkey IN TXT "g=; k=rsa; p=MHww[...]IDAQAB"
```

Codebeispiel 3.11: Resource Record-Beispiel: „DomainKeys“ (Schlüssel gekürzt)

Im Textteil stehen Tag-Value-Paare, deren mögliche Werte und Bedeutungen in Tabelle 3.2 aufgeführt sind. Aufgrund der bereits erwähnten Längenpro-

Tag	Bedeutung
g	Falls hier ein Wert angegeben ist, so muß er exakt mit dem Local-Part der Absenderadresse übereinstimmen. Mit diesem Tag kann bestimmt werden, welche Absenderadressen diesen Schlüssel benutzen dürfen (optional).
k	Gibt an, welcher Schlüsseltyp Verwendung findet (obligatorisch).
n	Feld für Kommentare (optional).
p	In diesem Tag wird der öffentliche Schlüssel als Base64-kodierte Zeichenkette angegeben. Ein leerer Inhalt bedeutet, daß der Schlüssel widerrufen wurde (obligatorisch).
t	Feld zur Angabe einiger Flags; bislang als möglicher Wert: ‘y’ bedeutet Testbetrieb (optional).

Tab. 3.2: Tag-Value-Paare für DomainKeys-Records

blematik bei TXT-RRs sollte insbesondere der Gebrauch des n-Tags sparsam erfolgen.

Um Namenskonflikte zu vermeiden, wird vorgeschlagen, für die öffentlichen Schlüssel jeweils die Subdomain `_domainkey.` zu reservieren und dort die Schlüssel gemäß obiger Spezifikation zu speichern. Dieser Namensraum wird durch die Einführung sog. Selektoren weiter unterteilt. Selektoren sind

[42] Siehe [16].

frei wählbare Namen und müssen im DNS-Namensraum und in E-Mail-Headern gültig sein. Damit wird die gleichzeitige Verwendung mehrerer Schlüssel innerhalb einer Domain ermöglicht[43].

3.3.3.2 Syntax des Signatur-Headers

Die Signatur einer E-Mail wird im „DomainKey-Signature“ genannten Header gespeichert. Dieser muß den bereits vorhandenen Headerzeilen vorangestellt werden:

```
DomainKey-Signature: a=rsa-sha1; q=dns;c=nofws;
   s=beta; d=gmail.com;h=received:message-id
   :date:from:reply-to:to:subject:mime-version
   :content-type:content-transfer-encoding;
   b=uaJjK+aijo2Dh0+L/iacte1t[...]kyVxvy7jW74eMFLSVTB2Q2hU=
```

Codebeispiel 3.12: Beispiel eines DomainKey-Signature-Headers (Signatur gekürzt)

Die möglichen Tag-Value-Paare haben dabei die in Tabelle 3.3 angegebene Bedeutung. Falls das h-Tag fehlt, werden alle auf den DomainKey-Signature-

Tag	Bedeutung
a	Signaturverfahren, bislang nur „rsa-sha1“ (obligatorisch).
b	Signatur (obligatorisch).
c	Kanonisierungsalgorithmus, siehe Seite 59 (obligatorisch).
d	Domainname der signierenden Domain (obligatorisch).
h	Durch Doppelpunkte getrennte Liste der Headernamen, welche zur Berechnung der Signatur verwendet wurden (optional).
q	Verfahren zum Zugriff auf den öffentlichen Schlüssel; bislang nur DNS vorgesehen (obligatorisch).
s	Selektor zur Auswahl eines DNS-Zweiges.

Tab. 3.3: Tag-Value-Paare des DomainKey-Signature-Headers

Header folgenden Kopfzeilen zur Berechnung der Signatur verwendet. Falls es vorhanden ist, werden die Kopfzeilen auf die aufgeführten beschränkt, wobei der From- bzw. der Sender-Header verwendet werden muß, und falls

[43] So könnte beispielsweise zu Beginn eines Jahres ein neuer Schlüssel unter dem Selektor `2006` gespeichert werden, während für eine Übergangszeit der alte noch unter `2005` verfügbar ist.

ein Header mehrmals vorkommt, z. B. Received, fließen alle Kopfzeilen dieses Typs ein.

3.3.3.3 Erstellen der Signatur

Die Signatur kann natürlich nur erstellt werden, wenn der private Schlüssel verfügbar ist und die entsprechenden Informationen über den Selektor und den öffentlichen Schlüssel vorliegen. Darüber hinaus darf eine Signatur *nicht* erstellt werden, falls sich der Absender nicht authentifiziert hat oder bereits ein DomainKey-Signature-Header vorhanden ist[44]. Die Signatur einer E-Mail wird nach folgendem Schema berechnet:

1. Die sendende Domain wird aus dem Sender-Header oder, falls dieser nicht vorhanden ist, aus dem From-Header ermittelt.

2. Auswahl eines privaten Schlüssels und des zugehörigen Selektors.

3. Unter Anwendung eines Kanonisierungsalgorithmus wird aus dem Mailbody und den Headerzeilen[45] ein gemeinsamer Hashwert errechnet und mit dem privaten Schlüssel verschlüsselt. Das Ergebnis ist die Signatur.

4. Der DomainKey-Signature-Header wird erstellt und der E-Mail vorangestellt.

3.3.3.4 Überprüfen der Signatur

Die Überprüfung einer E-Mail läuft ähnlich wie das Erstellen einer Signatur ab. Zunächst wird der DomainKey-Signature-Header gesucht und die darin enthaltenen Daten extrahiert. Anhand dieser Informationen wird zum einen der Hashwert der Mail errechnet und zum anderen ein DNS-Lookup nach

[44] Falls bereits eine Signatur vorhanden ist, aber ein weiterer Sender-Header hinzugekommen ist, der in der ersten Signatur noch nicht enthalten ist, darf eine neue Signatur erstellt werden. Die bereits vorhandene sollte nicht gelöscht werden.

[45] Entweder werden alle Headerzeilen verwendet oder nur die, welche durch das h-Tag spezifiziert werden, auf keinen Fall jedoch der DomainKey-Signature-Header selbst (siehe nächster Schritt).

`<SELEKTOR>._domainkey.<DOMAINNAME>` ausgeführt, dessen Antwort den öffentlichen Schlüssel enthält. Mit diesem wird die in der Mail enthaltene Signatur überprüft.

3.3.3.5 Eigenheiten von DomainKeys

Offensichtlich liegt bei erfolgloser Überprüfung – also auch wenn z. B. keine Signatur vorhanden ist – die weitere Vorgehensweise in den Händen des Empfängers. DomainKeys regt hierzu an, daß die sendende Domain ihre Signierungspolitik per DNS publizieren sollte. Von besonderem Interesse wäre, ob die Domäne das DomainKeys-Verfahren einsetzt, ob es nur getestet wird oder inwiefern stets alle ausgehenden Nachrichten signiert werden. Anhand solcher Aussagen könnte genauer ermittelt werden, wie mit unsignierten Nachrichten von dieser Domain verfahren werden soll. Zur Realisierung werden TXT-RRs vorgeschlagen, welche jenen ähneln, die der Publikation des öffentlichen Schlüssels dienen[46]. Allerdings werden diese direkt an dem Knoten `_domainkey` veröffentlicht.

Das Ergebnis der Überprüfung soll unter Verwendung der Headerzeile `Authentication-Results` gespeichert werden. Dieser neue Headername wird in dem Internet-Draft „Message Header for Indicating Sender Authentication Status“[47] von Murray Kucherawy beschrieben und soll die Ergebnisse beliebiger Authentifizierungsverfahren festhalten und Mailprogrammen zugänglich machen.

3.3.4 Identified Internet Mail (IIM)

Wie bereits erwähnt, gibt es zwischen DomainKeys und dem von einer Arbeitsgruppe der Firma Cisco entwickelten „Identified Internet Mail“-Verfahren zahlreiche Parallelen. Die jüngste, hier vorgestellte Fassung stammt vom Mai 2005[48]. Die wesentlichen Unterschiede liegen in der Schüsselverwaltung und in der Berechnung bzw. Prüfung der Signatur.

[46] Zur genauen Syntax der Tag-Value-Paare siehe [16] S.21ff.
[47] Siehe [53].
[48] Siehe [23].

Bei IIM kann die Korrektheit einer Signatur ohne irgendwelche Lookups überprüft werden, da in der IIM-Sig-Headerzeile alle hierfür benötigten Informationen enthalten sind, auch der öffentliche Schlüssel. Es fehlt lediglich eine Überprüfung, ob der Schlüssel tatsächlich von der angegebenen Domain stammt und in Verbindung mit der Absenderadresse verwendet werden darf. Für diesen Autorisierungsschritt können zwei alternative Verfahren eingesetzt werden. Die erste Möglichkeit besteht darin, daß für eine Domain ein Server zur Autorisierung der öffentlichen Schlüssel (sog. Key Registration Server, KRS) betrieben wird. Die zweite Möglichkeit sieht DNS-Lookups vor. Weder bei einer KRS- noch bei einer DNS-Abfrage wird der öffentliche Schlüssel übertragen, sondern lediglich sein Fingerabdruck und die E-Mail-Adresse des Absenders. Die Antwort liefert entweder eine Aussage, ob die Kombination aus Schlüssel und Adresse autorisiert ist, oder eine Adreßliste bzw. ein Vergleichsmuster (z. B. `*@example.com`). In diesem Fall muß von seiten des Empfänger noch überprüft werden, ob die Absenderadresse auf der Liste aufgeführt ist bzw. zu dem Muster paßt.

3.3.4.1 Syntax des Signatur-Headers

Wie folgendes Beispiel zeigt, wird die Signatur und weitere zu ihrer Berechnung nötigen Daten in Form von Tag-Value-Paaren in einem IIM-SIG-Header gespeichert:

```
IIM-SIG: v:"1"; h:"iim.example.com";
  d:"example.com"; z:"home"; m:"krs";
  t:"1094844765.338603"; x:"432000"; a:"rsa-sha1";
  b:"nofws:1192"; e:"Iw==";
  n:"zCnd+ByA23/7WMiIwaIZ7Ez[...]138IXLOvBVeaRZ4yWEPclZ/2Mda"
  "s5Bs9RPWH0BGd3fx6j+txdOXa[...]TexCOMFlDmatpXDXfFj3VI9o4G7"
  "674gFTasaoPcvEfZCwcBgZD7T[...]6sLZa3RTBUGzZqOshAMRpVek=";
  s:"Tg67/+k8oltzxIBxN4mevOg[...]ugZJ1VoaEm3bJ7JHAOy+X5FEMRF"
  "/SLZ+GBYIA7wtEmjgbHNuVRnb[...]bI6UKNGocCEX0TvVdZxFTQzbh3x"
  "zaEj6BIWx6GYIo8oWoeM3kzZT[...]iip2pPhuvaXu9Ho+3eR81MZ4=";
  c:"From: John Doe <jdoe@example.com>";
  c:"Date: Fri, 10 Sep 2004 12:25:51 -0700 (PDT)";
  c:"Subject: RE: Lunch"
```

Codebeispiel 3.13: Beispiel eines IIM-SIG-Headers (teilweise gekürzt)

Die verschiedenen Tags und ihre Bedeutung sind in Tabelle 3.4 aufgeführt. Wie aus obigem Beispiel ersichtlich und in der Tabelle erwähnt, kann das b-

Tag	Bedeutung
a	Verfahren zur Berechnung des Hashwerts und zur Verschlüsselung (obligatorisch).
b	Verwendeter Kanonisierungsalgorithmus mit optionaler Längenangabe bei `nofws` (optional).
c	Kopie eines zu verwendenden Headers; From bzw. Sender müssen, Subject und Date sollten aufgeführt sein. Weitere Header sind optional.
d	Domainname der signierenden Domain (obligatorisch).
e	Öffentlicher RSA-Exponent des zugrundeliegenden Schlüssels (obligatorisch).
h	Hostname des Rechners, welcher die Signatur für die E-Mail erzeugt hat (optional).
m	Methode, anhand der der Empfänger den öffentlichen Schlüssel überprüfen kann (optional).
n	Öffentlicher RSA-Modul des zugrunde liegenden Schlüssels (obligatorisch)
s	RSA-Signatur des berechneten Hashwerts (obligatorisch).
t	Zeitstempel, bestehend aus der Anzahl der Sekunden seit 1.1.1970, gefolgt von einigen Nachkommastellen, welche die Eindeutigkeit gewährleisten (obligatorisch).
v	Versionsnummer (obligatorisch).
x	Ablaufdatum, wann die Signatur ungültig wird, in Sekunden seit 1.1.1970 (obligatorisch).
z	Semantik der Signatur (optional, siehe Text).

Tab. 3.4: Tag-Value-Paare des IIM-Signature-Headers

Tag mit einer Längenangabe versehen sein. Der Integerwert besagt, wieviele Bytes des Body in die Berechnung der Signatur einfließen. Damit soll gewährleistet werden, daß die Signatur gültig bleibt, wenn – wie bei Mailinglisten häufig der Fall – an eine Nachricht bei einer Weiterleitung Informationen angefügt werden[49].

Das c-Tag bezeichnet einen Header, welcher in die Berechnung des Hashwertes einfließen soll. Anders als bei DomainKeys, wird hier jedoch nicht nur der Headername, sondern der ganze Header (Name und Inhalt) angegeben. Für das z-Tag stehen die Werte „home“ und „routing“ zur Verfügung, wobei

[49] Z. B. „Um von dieser Verteilerliste wieder herunter zu kommen...“.

mit dem ersten Wert ausgedrückt wird, daß die Signatur vom Ursprung der Nachricht hinzugefügt wurde, während „routing“ aussagt, daß die Signatur von einem Zwischensystem hinzugefügt wurde.

3.3.4.2 Erstellen und Überprüfen der Signatur

Unabhängig davon, ob die Signatur erstellt oder überprüft werden soll, die Berechnung erfolgt stets auf die gleiche Weise und ausschließlich unter Verwendung des IIM-SIG-Headers und des E-Mail-Textes: Zunächst wird für den Body ein Hashwert berechnet, wobei vorher eine Kanonisierung (siehe Seite 60) durchgeführt werden kann. Die Tags und Werte des IIM-SIG-Headers, mit Ausnahme des s-Tag, werden konkateniert und alle Syntaxelemente, wie Anführungsstriche, Strich- und Doppelpunkte, entfernt[50]. Aus diesen beiden Elementen – dem modifizierten Inhalt des IIM-SIG-Headers und dem Hashwert des Bodys – wird ein neuer Hashwert errechnet und anschließend signiert. Dieser Wert wird entweder unter Verwendung des s-Tag in den IIM-SIG-Header eingefügt (Erstellen der Signatur) oder mit dem Wert verglichen, welcher durch das s-Tag vorgegeben wird (Überprüfung der Signatur).

3.3.4.3 Überprüfung einer E-Mail

Zur Überprüfung einer E-Mail müssen zwei Sachverhalte kontrolliert werden. Zum einen wird anhand des im vorherigen Abschnitt beschriebenen Verfahrens die Konsistenz der Nachricht getestet, zum anderen muß die Autorisierung des öffentlichen Schlüssels geprüft werden, wofür zwei Verfahren möglich sind:

Autorisierung per KRS Der Empfänger einer signierten Mail überprüft, ob gültige Informationen im lokalen Cache vorhanden sind. Ist dies nicht der Fall, wird eine DNS-Abfrage nach einem KR-Record durchgeführt. Ein KR-Record ist ein neu geschaffener DNS-Textrecord, welcher Informationen zur Schlüsselverwaltung beinhaltet. In diesem Fall müßte der KR-Record

[50] Für den Header aus obigem Beispiel ergäbe sich `v1hiim.example.comdexample.comzhomemkrst1094844765.338603x432000arsa[...]Lunch`.

einen URI enthalten, welcher auf den zugehörigen Schlüsselserver verweist. An diesen wird dann eine HTTP-GET-Anfrage gestellt, welche neben dem Domainnamen die E-Mailadresse des Absenders und den Fingerabdruck des öffentlichen Schlüssels enthält. Der KRS überprüft nun anhand seines Datenbestands, ob die Kombination aus Schlüssel und Mailadresse autorisiert ist.

Autorisierung per DNS-Records Soll eine E-Mail anhand von DNS-Records überprüft werden, muss der Empfänger eine DNS-Anfrage nach einem KR-Record der Form `<keyfp>.<domain>` stellen, wobei `<keyfp>` den Fingerabdruck des Schlüssels bezeichnet. In diesem Fall liefert die Antwort direkt die Informationen, mit welchen Mailadressen der bezeichnete Schlüssel verwendet werden darf.

Ein KRS bietet eine größere Flexibilität bei der Schlüsselverwaltung, welche per DNS so nicht oder nur mit sehr großem Aufwand realisierbar ist. Es ist möglich, einem Schlüssel mehrere Adressen (Adreßliste) zuzuordnen oder einer Adresse einen oder mehrere Schlüssel. Bei der direkten Schlüsselverwaltung per DNS ist außerdem die Speicherung der E-Mailadresse im DNS-Record erforderlich, was im Hinblick auf den Datenschutz bedenklich sein könnte. Dieses Verfahren bietet sich vor allem für Domänen an, welche über eine geringe Anzahl an Mailadressen verfügen, weil in diesem Fall nur wenige oder sogar nur ein Schlüssel autorisiert werden müssen. Darüber hinaus kann der Betrieb eines KRS vermieden werden.

Der Inhalt eines KR-Records hat in beiden Fällen die gleiche Syntax und soll anhand der folgenden zwei Beispiele erläutert werden:

```
WDQGpekHKCmKyKWk._krs.example.com.     KR
    "v:IIM1; s:200; r:100; t:3600; m:*@example.com"

example.com.                          IN KR 10 10 378
    "v:IIM1; s:200; k:http://www.example.com/KRS/"
```

Codebeispiel 3.14: Resource Record-Beispiel: „IIM“

Der erste Record wird für eine DNS-Autorisierung verwendet, der zweite für eine Autorisierung per KRS. Eventuell notwendige Anführungsstriche und Backslashes zur Markierung derselben wurde aus Gründen der Lesbarkeit

weggelassen. Zur Bedeutung der Tags siehe Tabelle 3.5.

Tag	Bedeutung
a	Angabe zur Autorisierung, `fail`, `pass` oder `unknown` (optional).
c	Kommentar in Textform (optional).
k	URL des zu verwendenden KRS (optional).
m	Angabe eines Musters, auf das die E-Mailadresse passen muß (obligatorisch).
r	Angabe zur Güte, Werte zwischen -127 und 127, z. B. erhalten zurückgezogene Schlüssel einen negativen Wert; obligatorisch, falls kein a-Tag vorhanden ist.
s	Statuscode, analog zu den SMTP/HTTP-Statuscodes; zulässig sind: 200 (erfolgreiche Anfrage), 201 (erfolgreiche Anfrage, aber Kombination aus `keyfp` und Adresse nicht gefunden), 500 (permanenter Fehler).
t	Zeitangabe für die Cachingdauer in Sekunden (optional).
v	Versionsangabe (obligatorisch).

Tab. 3.5: Tag-Value-Paare für KR-Records

3.3.4.4 Eigenheiten von IIM

Falls eine Nachricht keine Signatur aufweist bzw. Fehler beim Überprüfen der Signatur aufgetreten sind, wird vorgeschlagen, einen sog. „Null Key Check“ durchzuführen. Hierzu wird aus dem d-Tag bzw., falls keine Signatur vorhanden ist, aus der Absenderadresse der Domainname extrahiert und ein DNS-Lookup nach einem KR-Record durchgeführt. Ein an entsprechender Stelle veröffentlichter Resource Record könnte dann Auskunft über die Richtlinien der Absenderdomain geben und anhand dieser die E-Mail abgelehnt oder angenommen werden.

Abschließend soll nicht unerwähnt bleiben, daß, ähnlich wie bei den bisher vorgestellten Verfahren, ein neuer Headertyp (IIM-Verify) definiert wird, in welchem das Ergebnis der Überprüfung festgehalten werden soll.

3.3.5 Domain Keys Identified Mail (DKIM)

Im Juni 2005 beschlossen Yahoo! und Cisco, ihre beiden Verfahren zu einem gemeinsamen Vorschlag zusammenzufassen und die Entwicklung unter der Bezeichnung „Domain Keys Identified Mail“ (DKIM, [68]) weiterzuführen. Seit November des gleichen Jahres werden die Bemühungen im Rahmen einer IETF-Arbeitsgruppe des gleichen Namens fortgesetzt. Laut Charta der Arbeitsgruppe[51] sollen bis Ende 2006 die Arbeiten so weit fortgeschritten sein, daß der Vorschlag als „Proposed Standard“ dem Standardisierungsprozeß unterzogen werden kann. Die folgenden Ausführungen beziehen sich auf die gegenwärtig aktuelle Fassung (August 2006) des als Internet-Draft veröffentlichten Vorschlags[52]. Aufgrund der hohen Aktivität der Arbeitsgruppe – es existiert bereits die vierte Version seit Februar 2006 – ist davon auszugehen, daß in unmittelbarer Zukunft noch weitere Versionen folgen werden, welche jedoch, ausgehend von der jüngsten Änderungshistorie, vermutlich nur Verbesserungen an Details beinhalten werden. Der wesentliche Unterschied zu seinen beiden Vorgängern besteht darin, daß DKIM zwischen dem Absender (bzw. Verfasser) einer E-Mail und dem Unterzeichner trennt (siehe d- und i-Tag des folgenden Abschnitts).

Wie seine beiden Vorgänger verwendet DKIM zur Speicherung der Daten Tag-Value-Paare. Der besseren Übersichtlichkeit wegen wurden in den Tabellen 3.6 und 3.7 einige Tags mit einem Subscript (*DK* bzw. *IIM*) ausgezeichnet. Damit soll veranschaulicht werden, von welchem Vorgänger das entsprechende Tag übernommen wurde. Hierbei ist zu beachten, daß sich in manchen Fällen die Bedeutung leicht geändert haben kann.

3.3.5.1 Syntax des Signatur-Headers

Der verwendete Header zur Speicherung der Signaturdaten ähnelt stark dem des DomainKeys-Verfahrens. Tabelle 3.6 gibt einen Überblick über die möglichen Tags und ihre Bedeutung.

[51] Siehe [67].
[52] Siehe [1].

Tag	Bedeutung
$a_{DK,IIM}$	Verwendeter Algorithmus zur Erzeugung der Signatur (obligatorisch). Prüfende Instanzen müssen `rsa-sha1` und `rsa-sha256` unterstützen, zur Erzeugung sollte `rsa-sha256` verwendet werden.
b_{DK}	Signatur (obligatorisch). Sie wird mittels Base64 kodiert, darin enthaltene FWS müssen ignoriert werden.
bh	Hashwert des normalisierten Mailbody (obligatorisch).
c_{DK}	Kanonisierungsverfahren für Header und Body (optional, default: `simple/simple`). Falls nur ein Verfahren angegeben ist, wird dieses für den Header verwendet (Beispiel: `c=relaxed` entspricht `c=relaxed/simple`).
$d_{DK,IIM}$	Domainname des Unterzeichners (obligatorisch). Anhand dieser Angabe wird nach dem öffentlichen Schlüssel gesucht. Diese Domain muß die gleiche oder eine übergeordnete Domain der Domain des i-Tags sein.
h_{DK}	Durch Doppelpunkte getrennte Liste der Headernamen, welche in die Berechnung der Signatur einfließen (obligatorisch). Der DKIM-Signature-Header darf nicht aufgeführt sein, er fließt immer in die Berechnung des Hashwertes ein (siehe 3.3.5.3).
i	Mailadresse, in deren Auftrag die Nachricht signiert wird (optional, default: '@', gefolgt von dem Domainnamen aus dem d-Tag). Der Local-Part der Mailadresse darf weggelassen werden, die Domain muß die gleiche oder eine Subdomain der Domain des d-Tags sein.
l	Längenangabe, wieviele Bytes des (normalisierten) Body in die Berechnung der Signatur einfließen (optional, default: gesamter Body).

Tag	Bedeutung
q_{DK}	Durch Doppelpunkte getrennte Liste von Verfahren zum Bezug des öffentlichen Schlüssels (optional, default: `dns/txt`). Jedes Verfahren wird nach dem Schema `Typ[/Optionen]` angegeben, wobei Syntax und Semantik der Optionen vom Typ abhängig sind. Derzeit nur `dns/txt`.
s_{DK}	Selektor zur Auswahl eines DNS-Zweiges (obligatorisch).
t_{IIM}	Zeitangabe, wann die Signatur erzeugt wurde (empfohlen, default: unbekannte Zeitangabe). Angabe erfolgt in Sekunden seit dem 1.1.1970, 0 Uhr UTC.
v_{IIM}	Versionsangabe (obligatorisch). Zur Zeit '0.4', entspricht der Version der DKIM-Spezifikation.
x_{IIM}	Zeitangabe, wann die Gültigkeit der Signatur erlischt (empfohlen, default: Gültigkeit erlischt nicht). Gleiches Format wie t-Tag.
z	Durch einen senkrechten Strich ('|') getrennte Liste ausgewählter Headernamen und -inhalte, welche zum Zeitpunkt der Signierung vorhanden waren. Dient ausschließlich der Fehlersuche und darf das Ergebnis einer Überprüfung nicht beeinflussen.

Tab. 3.6: Tag-Value-Paare des DKIM-Signature-Headers

Die Headernamen im h-Tag müssen in der gleichen Reihenfolge angegeben werden, wie sie bei der Berechnung der Signatur verwendet werden, und natürlich darf der DKIM-Signature-Header selbst nicht aufgeführt sein. Zusätzlich können in der Liste auch nicht vorhandene Header aufgeführt werden. Diese fließen als Zeichenketten der Länge Null in die Berechnung der Signatur ein. Dadurch wird erreicht, daß die Signatur ungültig wird, wenn zu einem späteren Zeitpunkt ein entsprechender Header dieses Namens zur Nachricht hinzugefügt wird. Beispielsweise verhindert die Angabe von „Sender“ oder „Resent-From“ etc. in gewissem Umfang eine Weiterleitung der Nachricht mit gültiger Signatur.

Für MIME-Nachrichten ist bei der Benutzung des l-Tag darauf zu achten, daß sich der signierte Bereich bis zum Abschluß der MIME-Sektion (gekennzeichnet durch `-CRLF`) erstreckt. Andernfalls könnte ein Angreifer weitere

MIME-Bestandteile einfügen, welche möglicherweise den signierten Nachrichtentext überdecken.

3.3.5.2 Bereitstellung des öffentlichen Schlüssels

Im Grunde übernimmt DKIM von beiden Vorgängern die Verfahren zur Schlüsselverwaltung, wobei der Einsatz eines Keyservers als Möglichkeit erwähnt, aber nicht näher beschrieben wird. Sofern für die Darstellung der Schlüsseldaten kein strukturiertes Format (z. B. XML) verwendet wird, ist zur textuellen Darstellung das in Tabelle 3.7 beschriebene System aus Tag-Value-Paaren zwingend zu verwenden. Insbesondere gilt dies bei der Verwendung von DNS-Resource-Records, welche bislang als einzige Möglichkeit zur Bereitstellung des Schlüssels genauer spezifiziert ist.

Analog zu DomainKeys werden die Schlüssel als TXT-Records in der reservierten Subdomain `_domainkey` gespeichert, welche durch Selektoren weiter untergliedert wird. Ein Lookup nach einem öffentlichen Schlüssel läuft dementsprechend genauso ab, wie bei DomainKeys (Seite 63) beschrieben.

Tag	Bedeutung
g_{DK}	Granularität (optional, default: '*'). Dieser Wert muß mit dem Local-Part der im i-Tag des Signatur-Headers (siehe Seite 71) angegebenen Adresse übereinstimmen. Damit kann spezifiziert werden, welche Adresse(n) den betreffenden Selektor verwenden dürfen (Wildcarding).
h	Durch Doppelpunkte getrennte Liste möglicher Hash-Algorithmen (optional, default: alle).
k_{DK}	Angabe des verwendeten Schlüsseltyps (optional, default: 'rsa').
n_{DK}	Feld für Kommentare (optional, default: leer).
p_{DK}	Öffentlicher Schlüssel als Base64-kodierte Zeichenkette (obligatorisch). Ein leerer Inhalt bedeutet, daß der Schlüssel widerrufen wurde.
s	Durch Doppelpunkte getrennte Liste möglicher Servicetypen (optional, default: '*'). Zur Zeit spezifizierte Angaben: `*, email`.

Tag	Bedeutung
t_{DK}	Durch Doppelpunkte getrennte Liste von Flags (optional, default: keine Flags). Zur Zeit spezifizierte Angaben: y (Testbetrieb), s (der Domainname des i-Tags aus dem Signatur-Header muß mit dem Domainname aus dem d-Tag übereinstimmen).
v_{IIM}	Versionsangabe (empfohlen, default: 'DKIM1'). Falls vorhanden, muß dieses Tag als erstes aufgeführt und sein Wert „DKIM1“ sein.

Tab. 3.7: Tag-Value-Paare für DKIM-DNS-Records

3.3.5.3 Berechnung der Hashwerte

Im ersten Schritt wird für den kanonisierten Mailbody – eventuell unter Beachtung einer Längenangabe (l-Tag) – ein Hashwert errechnet und mittels Base64 kodiert. Dieser Wert wird später beim Erstellen der Signatur im bh-Tag des Signatur-Headers aufgeführt. Im nächsten Schritt wird der Hashwert der Headerzeilen berechnet. Hierzu werden alle Headerzeilen, welche im h-Tag aufgeführt sind, mit ihrem abschließenden Zeilenende (CRLF) konkateniert und normalisiert. An die entstehende Zeichenkette wird der gesamte, kanonisierte DKIM-Signature-Header angehängt, wobei allerdings der Inhalt des b-Tags (die Signatur) leer bleibt. Da im bh-Tag bereits der Hashwert des Mailbody gespeichert ist, fließt dieser hierbei in die Berechnung ein. Falls die Nachricht in kodierter Form (Quoted-printable oder Base64) verschickt werden soll, muß diese vor der Berechnung der Hashwerte angewendet werden.

3.3.5.4 Erstellen und Überprüfen der Signatur

Eine Nachricht muß signiert sein, bevor sie den Verwaltungsbereich einer Organisation verläßt. Bevor hierfür die Hashwerte berechnet werden können, müssen die dabei zu verwendenden Headerzeilen ausgewählt und deren Namen im h-Tag aufgeführt werden. Die aktuelle Spezifikation macht hierzu wenige Vorgaben: Der From-Header muß signiert werden, und veränderbare Header (Return-Path) sollten nicht signiert werden. Zusätzlich können nicht vorhandene Header aufgeführt werden, um deren Fehlen explizit aus-

zudrücken[53]. Falls mehrfach vorkommende Header (Received) in die Signatur aufgenommen werden sollen, müssen diese mehrfach aufgeführt werden. Hierbei wird stets der physikalisch letzte dieses Namens verwendet. Sind beispielsweise folgende Received-Header vorhanden

```
Received: A
Received: B
Received: C
```

werden mit `c=Received:Received` die Header C und B verwendet. Es wird empfohlen, zusätzlich zumindest Date, Subject, Reply-To, Sender und alle MIME-Header zu signieren.

Wenn alle Header ausgewählt sind und die Nachricht eventuell zusätzlich kodiert wurde, wird gemäß obigem Verfahren der Hashwert berechnet und mit einem geeigneten privaten Schlüssel verschlüsselt. Das Ergebnis wird Base64-kodiert und in den Signatur-Header eingefügt (b-Tag).

Um die Signatur einer E-Mail zu überprüfen, werden der DKIM-Signature-Header analysiert und anhand der darin festgehaltenen Parameter wie in 3.3.5.3 beschrieben zwei Hashwerte berechnet. Zunächst wird der Hashwert des Mailbody mit dem im bh-Tag gespeicherten verglichen. Nur wenn diese übereinstimmen, wird anhand der im q-Tag spezifizierten Methode der öffentliche Schlüssel bezogen. Bevor mit diesem die Signatur aus dem b-Tag entschlüsselt und mit dem zuvor berechneten verglichen werden kann, ist zu prüfen, ob die Ergebnisse des Lookups mit den Parametern des Signatur-Headers in Einklang stehen. Insbesondere sind eventuelle Einschränkungen durch das g-Tag zu beachten und mit der im i-Tag gespeicherten Adresse zu vergleichen.

Prinzipiell kann diese Überprüfung zu einem beliebigen Zeitpunkt durchgeführt werden, allerdings sollte sie möglichst zeitnah bereits bei Empfang einer Nachricht durch einen Border-MTA erfolgen, da der verwendete Schlüssel später seine Gültigkeit verlieren kann. Das Ergebnis wird wie bei DomainKeys in einer Authentication-Results-Headerzeile (siehe Seite 64) gespeichert und ist somit – zumindest wenn der Schlüssel nicht gelöscht, sondern nur nicht mehr gültig ist – noch nachvollziehbar.

[53] Falls ein solcher Header später hinzugefügt wird, schlägt die Überprüfung der Signatur fehl.

Im Rahmen der DKIM-Arbeitsgruppe soll ebenfalls ein Protokoll erstellt werden, welches es einem Domaininhaber ermöglicht, sein Signierungsverhalten (sender signing policy) zu veröffentlichen. An diesen Richtlinien kann sich ein Empfänger bei seiner weiteren Vorgehensweise orientieren, wenn die Überprüfung einer Nachricht fehlschlägt. Leider sind in diesem Bereich noch keine Ergebnisse vorzuweisen, bislang ist die Arbeitsgruppe noch damit beschäftigt, die Anforderungen an diesen Mechanismus zu diskutieren.

3.3.6 Diskussion

Die hier aufgeführten Signaturverfahren haben den Nachteil, daß signierte Nachrichten, sofern sie nicht verändert werden, erneut verschickt werden können und dabei die Signatur ihre Gültigkeit behält. Dies läßt sich, wie in [1] beschrieben, für einen Replay-Angriff nutzen, welcher beispielsweise wie folgt Auswirkungen auf die Reputation des Opfers haben könnte: Ein Angreifer erhält von seinem Opfer eine signierte E-Mail, welche von einem unbeteiligten Dritten als Spam aufgefaßt werden kann (beispielsweise ein auf Anfrage erstelltes Verkaufsangebot). Diese Nachricht sendet er unverändert und möglicherweise wiederholt an sehr viele Empfänger. Die Wahrscheinlichkeit ist groß (vor allem bei Wiederholung), daß einige Empfänger die Nachricht als Spam auffassen und sich somit die Reputation des Opfers verschlechtert. Diese Gefahr läßt sich jedoch für das Opfer minimieren, wenn darauf geachtet wird, daß To- und CC-Header signiert werden. In diesem Fall kann erkannt werden, daß die E-Mail ursprünglich an eine andere Adresse gerichtet war.

Durch die Trennung von Verfasser und Unterzeichner bei DKIM entsteht für den Empfänger ein Problem, wenn eine Nachricht zwar eine gültige Signatur aufweist, aber die im i-Tag angegebene Identität nicht im Einklang mit der Adresse des Absenders (From) steht. In diesem Fall ist die Nachricht nach weiteren Absenderangaben zu durchsuchen und zu prüfen, ob diese akzeptiert werden können (Sender- oder Resent-Header), oder es muß untersucht werden, ob durch den Inhaber der From-Domain Richtlinien zu seinem Signierungsverhalten veröffentlicht wurden.

Insgesamt ist DKIM ein Verfahren, welches sowohl vom Absender als

auch vom Empfänger eine gewisse Sorgfalt erfordert. Dafür bietet es wichtige Vorteile: Zum einen erstreckt sich das direkte Vertrauensverhältnis nicht nur auf die beiden Partner einer SMTP-Session, sondern spannt einen Bogen vom Unterzeichner einer Nachricht bis zur Überprüfung desselben. Zum anderen wird die Integrität einer E-Mail gewährleistet, wodurch diese auch in rechtlichen Belangen einen höheren Stellenwert erhält.

Letztendlich könnte DKIM nicht zur dazu verwendet werden, Signaturen von erhaltenen Nachrichten zu überprüfen, sondern es könnte sogar zur Verschlüsselung von zu versendenden Nachrichten dienen. Da DKIM bereits die Möglichkeiten zur Schlüsselverwaltung bietet, fehlt im Grunde lediglich ein Mechanismus, anhand dessen ein zu einer E-Mailadresse gehörender öffentlicher Schlüssel bezogen werden könnte. Dies ist gemäß der aktuellen Spezifikation nicht möglich, da einerseits für einen Lookup Selektoren benötigt werden und zum anderen DKIM zwar im Prinzip für jede Mailadresse Schlüssel verwalten kann, de facto aber von dieser Möglichkeit jedoch kaum Gebrauch gemacht werden wird, da damit ein erheblicher Verwaltungsaufwand verbunden ist.

3.4 Kollaborative Verfahren

Spam-Versender machen sich natürlich nicht die Mühe, jede E-Mail in einem eigenen Vorgang zu erstellen und zu versenden, sondern ein und die selbe Nachricht wird tausendfach verschickt. Dies bedeutet, daß sich die weltweit versandten Spam-Mails in vergleichsweise wenige Klassen „identischer“ Nachrichten einteilen lassen. Kollaborative Verfahren versuchen diese Tatsache auszunutzen, indem für eingehende Nachrichten ein Fingerprint erstellt und auf einem zentralen Server nachgesehen wird, ob dieser vorhanden ist. Durch die Verwendung eines Fingerprints wird die Privatsphäre des Mail-Empfängers geschützt und der Ressourcenverbrauch (Bandbreite, Speicherplatz) gering gehalten. Falls der Fingerprint bereits bekannt ist, handelt es sich bei der Nachricht um eine Spam-Mail, welche zu einem früheren Zeitpunkt von einem anderen Teilnehmer empfangen und als Spam klassifiziert wurde. Falls der Fingerprint noch nicht bekannt ist und es sich um eine Spam-

Mail handelt, wird dieser auf dem Server gespeichert. Theoretisch reicht es also aus, *eine* Nachricht einer bestimmten Spam-Klasse durch *einen* Empfänger zu klassifizieren, statt jede Nachricht dieser Klasse durch jeden Empfänger. In der Praxis gestaltet sich das Verfahren jedoch komplexer und weniger effizient. Zum einen muß mit bewußten oder irrtümlichen Falschaussagen gerechnet werden, zum anderen werden viele Nachrichten einer bislang unbekannten Spam-Klasse ungefähr zur gleichen Zeit von mehreren Teilnehmern empfangen und klassifiziert werden. Letzteres ist jedoch nicht unwillkommen, weil sich dadurch die Konfidenz der Klassifizierung erhöht.

Eine erste Implementierung dieses Ansatzes erfolgte 1998 durch Vipul Prakash und ist als Sourceforge-Projekt unter dem Namen „Vipul's Razor" [78] bekannt. Die aktuelle Weiterentwicklung – „Cloudmark Network Classifier (CNC)" genannt – wird von Cloudmark in verschiedenen Produkten[54] kommerziell eingesetzt. Im folgenden soll das von Prakash in [79] beschriebene Verfahren kurz vorgestellt werden.

Die zentralen Bestandteile von CNC bestehen aus einem Client auf Seiten des Mail-Empfängers, einem Katalog-Server, einem Nominierungsserver und einem Trust-System. Falls für eine Nachricht kein Fingerprint auf dem Katalog-Server gefunden wird oder diese fälschlicherweise als Spam klassifiziert wurde, wird deren Fingerprint an den Nominierungsserver geschickt. Wird der gleiche Fingerprint ebenfalls von anderen Teilnehmern eingereicht gilt er als bestätigt. Aufgrund der Reputation der einreichenden Teilnehmer entscheidet das Trust-System welche Fingerprints auf den Katalog-Server verschoben, bzw. von diesem gelöscht werden müssen.

Weiter oben wurde erwähnt, daß ein Fingerprint „identische" Nachrichten zu einer Klasse zusammenfaßt. Die Schwierigkeit hierbei ist, einen Algorithmus zu finden, welcher eine Spam-Mail zuverlässig einer Spam-Klasse zuordnet und somit für dieses Verfahren den Gleichheitsbegriff definiert. Kryptographische Hashverfahren reagieren empfindlich auf Veränderungen und sind offensichtlich wenig geeignet. Der Algorithmus muß einerseits die Charakteristik einer Nachricht genau erfassen, aber andererseits unempfindlich gegenüber Veränderungen sein, wie beispielsweise geänderte Absenderanga-

[54] Siehe [12].

ben, zufällige Zeichenketten im Mailbody etc. Ist die Differenzierung zu stark oder zu schwach erhöht dies die Zahl der False Negatives bzw. False Positives. Der größte Schwachpunkt ist jedoch die Verwendung zentraler Server, welche anfällig für Ausfälle sind und ein willkommenes Ziel für Angriffe jeglicher Art darstellen (Single Point of Failure/Attack).

Verteilte kollaborative Verfahren Um den Single Point of Failure der einfachen kollaborativen Verfahren zu umgehen, gibt es verschiedene Ansätze, die Datenbasis auf mehrere Instanzen zu verteilen. Von Rhyolite Software wird eine Implementierung namens „Distributed Checksum Clearinghouse (DCC)“ bereitgestellt[55], die die Verwendung mehrerer Server vorsieht, welche in unregelmäßigen Abständen die am häufigsten vorkommenden Fingerprints austauschen. Andere Verfahren sehen den Einsatz von Peer-to-Peer-Netzwerken oder Multi-Agentensystemen vor.[56] Deren grundlegende Funktionsweise entspricht dem zuvor erwähnten Verfahren mit zentralem Server. Allerdings erweitert um die Funktionalität, Informationen an andere Agenten (bzw. Peers) weiterzureichen.

Der Vorteil der schwierigeren Angreifbarkeit durch den Einsatz verteilter Systeme wird durch den Nachteil eines relativ hohen Kommunikationsaufwands erkauft. Zum einen kann es sein, daß eine überprüfende Instanz über keine Informationen bezüglich einer bestimmten E-Mail verfügt und deswegen einen rekursiven Lookup-Prozess in Gang setzen muß, um die entsprechende Information zu erhalten, zum anderen müssen die Änderungen und neuen Klassifizierungen im Netzwerk bekannt gemacht werden. Je nach Verfahren sind diese beiden Vorgänge unterschiedlich aufwendig; je mehr Aufwand bei dem einen getrieben wird, desto weniger ist bei dem anderen nötig.

Über die Erfolgsquoten der implementierten Verfahren liegen nur wenige Ergebnisse vor. Für das Produkt SpamNet gibt der Hersteller Cloudmark eine Erkennungsquote von 75% an, manche Nutzer sprechen auch von 90%. Für Vipul's Razor konnten keine Zahlen ermittelt werden, was unter Umständen daran liegt, daß es häufig in Kombination mit anderen Verfahren eingesetzt

[55] Siehe [84].
[56] Siehe z. B. [13,25] bzw. [51,54].

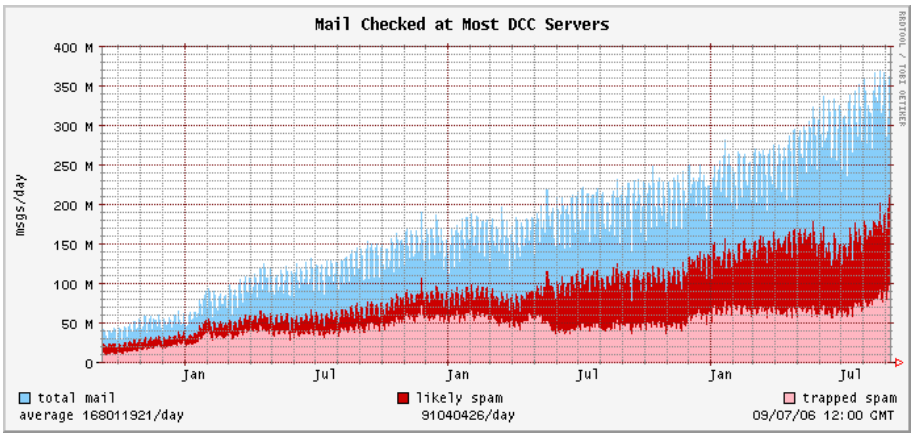

Abb. 3.1: DCC Drei-Jahres-Statistik: Ham- und Spam-Aufkommen [85]

wird. Lediglich von Rhyolite Software werden detailliertere Graphiken veröffentlicht. Abbildung 3.1 zeigt die absolute Entwicklung der letzten drei Jahre. Während die Gesamtzahl der E-Mails und die Anzahl der vermutlich als Spam erkannten Nachrichten deutlich angestiegen ist, hat sich das Aufkommen der als Spam erkannten Nachrichten langsamer entwickelt. De facto bedeutet dies also eine Verschlechterung bei der Erkennung von Spam, was bei der Betrachtung relativer Zahlen (Abbildung 3.2) deutlich zu erkennen ist. Des weiteren ist daraus zu entnehmen, daß der Anteil der vermutlichen

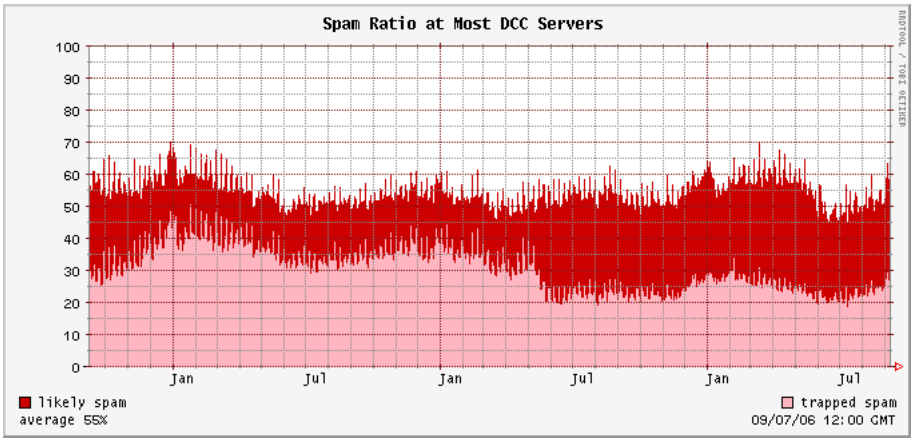

Abb. 3.2: DCC Drei-Jahres-Statistik: Relatives Spam-Aufkommen [85]

Spam-Mails (likely spam, also jene Nachrichten, welche einer genaueren Über-

prüfung bedürfen) mit ca. 55% relativ hoch ist. Das bedeutet, daß knapp nur jede zweite E-Mail korrekt klassifiziert wurde. Daß sich die Erkennungsrate verschlechtert hat, könnte daran liegen, daß es Spamversendern zunehmend gelingt ihre Nachrichten so zu gestalten, daß die Kategorisierung der eintreffenden E-Mails erschwert wird und es somit häufiger vorkommt, daß ein bestimmter Fingerprint noch nicht im Datenbestand gespeichert ist.

3.5 Camram

Camram[57] – abgeleitet von „Campaign for Real Mail“ – beschreibt ein Verfahren, welches nach dem Sender-Pays-Prinzip funktioniert, allerdings erfolgt die Bezahlung nicht monetär, sondern in Form von Rechenzeit. Das System wirkt sich sowohl auf ein- als auch auf ausgehende E-Mails wie folgt aus: Ausgehende Nachrichten werden mit einer digitalen Briefmarke versehen[58] und die Adresse an die diese E-Mail gerichtet ist, wird in einer Whitelist gespeichert, welche für die Analyse der eingehenden E-Mails verwendet wird.

Eingehende Nachrichten werden einem mehrstufigen Test unterzogen:

1. Zunächst wird geprüft, ob für die eingehende Nachricht eine Bezahlung erfolgt ist. Falls das zutrifft, wird die Nachricht in den Eingangsordner (Inbox) verschoben und die Absenderadresse zur Whitelist hinzugefügt.

2. Im nächsten Schritt wird die Absenderadresse mit der Whitelist abgeglichen. Findet sich der Absender in dieser Liste, wird die E-Mail ebenfalls in den Eingangsordner verschoben.

3. Wenn die E-Mail eine Antwort auf eine früher verschickte Aufforderung zur Nachgebühr[59] ist, wird die Nachgebühr geprüft und die zugehörige Nachricht aus der Quarantäne (siehe 4b) entlassen und in die Inbox verschoben.

4. Zuletzt durchläuft die E-Mail einen herkömmlichen Spamfilter, welcher unter folgenden drei Kategorien differenziert:

[57] Siehe [50].
[58] Sozusagen die Bezahlung oder das Porto für die E-Mail.
[59] Eng.: “postage-due notice”, siehe 4c.

(a) Echte E-Mails („Ham“) werden in die Inbox verschoben.

(b) E-Mails, deren Status nicht eindeutig zu entscheiden ist („Mystery Meat“), werden unter Quarantäne gestellt; für diese wird an den Absender eine Aufforderung zur Zahlung einer Nachgebühr verschickt.

(c) Für Spam werden unterschiedliche Behandlungsweisen vorgeschlagen. Entweder sie werden verworfen, oder sie werden unter Quarantäne gestellt und ignoriert, oder sie werden unter Quarantäne gestellt und für sie wird eine Aufforderung zur Nachgebühr verschickt.

Eine Aufforderung zur Nachgebühr (postage-due notice) ist eine E-Mail an den Absender, welche einen Hyperlink zu einem Java-Programm enthält. Mit diesem wird nachträglich eine Briefmarke für eine frühere E-Mail erzeugt und an den Empfänger per E-Mail geschickt.

Die Whitelist ermöglicht die Umgehung des Camram-Systems für Absender, von denen kein Spam zu erwarten ist. Hierzu werden in diese Liste automatisch jene Mailadressen eingetragen, an die der Empfänger selbst E-Mails geschickt hat. Als weitere Quellen für solche Adressen könnten Antworten auf die Aufforderungen zur Nachgebühr, existierende Adreßbücher oder die Ergebnisse des Spamfilters ausgewertet werden.

Ein von den Autoren erwähnter Nachteil der Whitelist besteht darin, daß sie nicht in die Hände von Spammern fallen darf. Damit stünde diesen eine Liste validierter E-Mailadressen zur Verfügung; die Adressen könnten benutzt werden, um Spam an den Besitzer der Whitelist oder unter dessen Namen Spam an die Adressen auf der Liste zu senden. Als Lösung hierfür wird ein dezentralisiertes, auf OpenPGP beruhendes Kryptoverfahren vorgeschlagen. Hierbei würden in der Whitelist nicht mehr Mailadressen, sondern öffentliche Schlüssel gesammelt. Jede Nachricht muß signiert werden, und das Camram-System wäre damit in der Lage, vertrauenswürdige Absender nicht mehr anhand der Mailadresse, sondern anhand der Signatur zu erkennen. Damit entspräche dieses System in Grundzügen dem DKIM-Verfahren mit User-basierten Schlüsseln, und das Erstellen von Briefmarken könnte komplett

entfallen.

Bei der Erzeugung der digitalen Briefmarken ist Camram flexibel, und es können verschiedene Verfahren zum Einsatz kommen. Die Autoren Johansson und Dawson schlagen den von Adam Back entwickelten Hashcash-Algorithmus[60] vor. Dieser führt anhand verschiedener Eingangswerte (Zufallszahl, Datum, Uhrzeit, Absender- und Empfängeradresse) die Berechnung eines Tokens durch. Dabei wird ein signifikanter Aufwand an Zeit und Rechenleistung generiert, welcher durch interne Parameter gesteuert werden kann. Die Überprüfung des Tokens, welches als digitale Briefmarke (bzw. proof-of-work stamp) dient, ist wiederum einfach und ohne rechenintensiven Aufwand möglich.

Die Problematik, inwiefern die mit der Zeit steigende Rechenleistung die digitalen Briefmarken wertlos macht, wird ebenfalls in dem Artikel aufgeführt. Auch wenn der Berechnungsaufwand bei Hashcash parametrisierbar ist, wird befürchtet, daß eines Tages das Mooresche Gesetz die Grundlage hierfür entzieht. Als Lösungsmöglichkeit wird darauf hingewiesen, daß eine Technik gefunden werden müßte, welche weniger den Prozessor, als vielmehr beispielsweise den Speicherbus belastet, weil hier die jährlichen Geschwindigkeitszuwächse deutlich moderater ausfallen. Dennoch bleibt festzuhalten, daß auch hier die Leistungen exponentiell wachsen.

Recht ausführlich widmen sich die Entwickler von Camram dem Aspekt, daß das Erzeugen entsprechender Briefmarken durch sog. Zombie-PCs[61] dieses System nutzlos machen würde[62]. Anhand einer Beispielrechnung mit plausiblen Zahlen wird begründet, daß das Verfahren genügend Möglichkeiten böte, die Berechnung einer digitalen Briefmarke so aufwendig zu machen, daß Zombie-PCs keine Gefahr darstellten, aber der normale Benutzer nicht beeinträchtigt würde. Bei 3 Mio. Zombie-PCs und 13,1 Mrd. zu verschickende Spam-Mails müßte die Berechnung einer digitalen Briefmarke zwischen einer

[60] Siehe [6].

[61] Zombie-PCs sind Computer, welche Dritten unter Ausnutzung von Schwachstellen den Zugriff von außerhalb ermöglichen. Diese Rechner werden oft verwendet, um Spam, Viren oder Würmer zu verbreiten oder verteilte Angriffe (DDOS) zu starten. Ein Netz aus solchen Zombie-PCs wird Bot-Netz genannt.

[62] Siehe [70].

und drei Minuten benötigen.

Darüber hinaus ergeben sich noch einige weitere Nachteile, welche hier stichpunktartig aufgezählt seien:

- Durch Erzeugung einer digitalen Briefmarke ist die Zustellung einer Spam-E-Mail garantiert. Durch das Whitelisting ist es sogar ausreichend, dies für jede Absenderadresse nur einmal durchzuführen.
- Für große Firmen mit vielen ausgehenden E-Mails führt Camram zu einer deutlichen Belastung für den SMTP-Server. Andernfalls muß die Software auf jedem Firmen-PC installiert sein.
- Solange dieses Verfahren keine weite Verbreitung erreicht, ist der Nutzen eher gering, und letztendlich entscheidet doch wieder ein Spamfilter über die Trennschärfe.
- Belästigung unbeteiligter Dritter durch gefälschte Absenderadressen.
- Postage-due-Nachrichten führen zur Validierung von Mailadressen.

Es bleibt festzuhalten, daß das Camram-System zwar einige recht interessante Ansätze und Ideen liefert, aber in einigen Bereichen (z. B. Umgang mit Mailinglisten, postage-due-Nachricht für jede Spam-Mail, Einsatz in Firmen) noch nachgebessert werden müßte. Letztendlich ist es zu den Challenge-Response-Verfahren zu zählen. Die diesen Verfahren immanente Problematik wird im folgenden Abschnitt dargelegt.

3.6 Challenge-Response-Verfahren

Challenge-Response-Verfahren (CR) versuchen die Autorisierung einer Adresse dadurch zu überprüfen, daß auf eine Nachricht von einem unbekannten Absender mit einer E-Mail (Challenge) geantwortet wird, in der um eine Bestätigung (Response) der Absenderadresse oder um eine nochmalige Zusendung der Mail gebeten wird. Manchmal wird in der Challenge-Nachricht auch das Lösen einer zusätzlichen Aufgabe gefordert.

Verfahren dieser Art scheitern aus mehreren Gründen. Um einen wirklichen Beitrag zur Autorisierung des Absenders zu leisten, müßte *jede* Nachricht anhand dieses Verfahrens überprüft werden, nicht nur die von unbekannten Absendern, was eine Verdreifachung des E-Mailaufkommens zur Folge hätte. Wird nur die erste Nachricht eines Unbekannten überprüft und alle weiteren angenommen, ist es ausreichend eine einzige dem Empfänger bekannte Adresse zu benutzen, um ungehindert Spam an diesen zu verschicken. Die Ermittlung solcher Adressen ist bei weitem einfacher, als es zunächst scheinen mag: Bei größeren Firmen und Organisationen dürfte der Ansatz erfolgversprechend sein, Adressen aus der gleichen Domain, aber mit anderem Local-Part auszuprobieren[63]. Bei privaten Adressen bietet die Analyse von Foren, Newsgruppen oder Mailinglisten eine gute Chance, Bekanntheitsbeziehungen offenzulegen und entsprechende Adressen auszuprobieren.

CR-Systeme sind aber vor allem deswegen abzulehnen, weil sie zu nicht geringem Schaden führen können, wenn eine Absenderadresse mißbräuchlich verwendet wird. Im besten Fall, nämlich dann, wenn diese Adresse tatsächlich existiert, wird die Challenge-Nachricht an einen Unbeteiligten Dritten geschickt, was dort zu entsprechender Belästigung führt. Damit hat die Challenge-Nachricht den einer Spam-Mail vergleichbaren Effekt, der Betreiber des CR-Systems wird quasi selbst zum Spam-Versender. In dem Fall, daß die mißbräuchlich verwendete Adresse nicht existiert, wird die Challenge-Nachricht mit einem Bounce beantwortet, was möglicherweise zu einer Endlosschleife aus Bounce- und Challenge-Nachrichten führt[64].

Ein anderer, einem CR-System ähnlicher Ansatz bestünde darin, ein Protokoll zu entwickeln, welches anhand der Received-Header den Weg einer E-Mail (bzw. den MSA) überprüft und so die tatsächliche Senderdomain ermittelt. Vorbedingung hierfür wäre, daß jede am Sendevorgang beteiligte Instanz einen Hashwert für die Nachricht speichert. Der Empfänger könnte aus dem Received-Header – beginnend bei dem historisch ersten (also dem physikalisch letzten) – die Adresse des sendenden MTA ermitteln und dort anfragen,

[63] Oftmals wird bei Firmen auch der Local-Part nach einem festen Schema gebildet, z. B. die ersten vier Buchstaben des Vornamens und die ersten drei des Nachnamens etc.

[64] Eine Endlosschleife droht auch, wenn der Empfänger der Challenge-Nachricht seinerseits ein CR-System betreibt.

ob eine E-Mail mit dem entsprechenden Hashwert verschickt wurde. Ein Vereinfachung wäre, daß der MSA ermittelt wird und nur bei diesem angefragt wird.

Dieser Ansatz scheitert aus zwei Gründen. Zum einen sind die in einem Received-Header gemachten Angaben oft nicht vollständig bzw. können gefälscht sein, zum anderen ist der zu betreibende technische Aufwand im Vergleich zum erzielten Vorteil zu groß. Allein die Hashdaten erreichen bei großen Providern mit mehreren Millionen Mails pro Tag schnell einige hundert Megabytes[65]. Zur Verwaltung dieser Daten und zur Bewältigung der Anfragen wäre ein sehr leistungsfähiger Datenbankserver erforderlich, was mit erheblichen Kosten verbunden ist, woraus der Betreiber des MSA keinen direkten Nutzen ziehen kann. Die Vergangenheit zeigt jedoch, daß sich neue Systeme nur dann durchsetzen können, wenn ein erhöhter Aufwand zu einem direkten Nutzen führt.

3.7 Zusammenfassung

Sowohl SPF/Sender ID als auch DKIM sind Verfahren, welche die verlässliche Überprüfung der Absenderdomain ermöglichen. Dadurch erhält der Empfänger ein Instrument, welches das Ungleichgewicht zwischen ihm und dem Absender aufhebt, denn hiermit kann eine E-Mail auf User-Ebene zurückgewiesen werden, was ohne diesen Hilfsmitteln nur auf Ebene des MDA möglich ist. Im Grunde wäre sogar die Autorisierung einzelner Benutzer möglich, allerdings ist zu erwarten, daß von dieser Möglichkeit eher kaum Gebrauch gemacht werden wird, da dies erstens mit einem beträchtlichem Konfigurationsaufwand einhergeht, zweitens ist das Domain Name System zu statisch, um den häufigen Änderungen einer Benutzerverwaltung gerecht zu werden, und drittens würde die Anzahl der DNS-Anfragen deutlich ansteigen, weil Caching-Effekte wegfallen. Andererseits ist eine Überprüfung der Absenderdomain ausreichend, wenn von dieser garantiert *und* publiziert wird, daß sich der Benutzer gegenüber der Domäne authentifiziert hat (SMTP-AUTH,

[65] Länge einiger Hashwerte: 20 Bytes (SHA1), 32 Bytes (SHA256), 48 Bytes (SHA384)

„POP3 vor SMTP“ oder ein gleichwertiges Verfahren).

Bei allen drei Verfahren ist der Mailversand mit einem erhöhten Aufwand verbunden. Zu dem üblichen MX-Lookup kommt mindestens ein weiterer DNS-Lookup durch den empfangenden SMTP-Server hinzu, d. h. die Anzahl der benötigten DNS-Abfragen wird verdoppelt. Je nach Verfahren sind eventuell weitere Lookups notwendig, die Antworten müssen teilweise aufwendig geparst werden, und im Falle von DKIM sind teure kryptographische Berechnungen durchzuführen; dies alles führt zu einem weiteren Anstieg der Rechenlast. Es gibt jedoch Anzeichen, daß der erhöhte Aufwand auf seiten der DNS- und SMTP-Server möglicherweise dadurch überkompensiert wird, daß deutlich weniger E-Mails in das Mailsystem gelangen[66] und somit die Gesamtbilanz positiv ausfällt, zumindest solange Spammer gefälschte Absenderadressen verwenden.

Es wäre begrüßenswert, wenn beide Ansätze in Zukunft in einem gemeinsamen Protokoll zusammengeführt würden, da sich die Vorteile von SPF/-Sender ID einerseits und DKIM andererseits gegenseitig ergänzen: Zur Authentifizierung eines Absenders während einer SMTP-Sitzung ist SPF/Sender ID besser geeignet, da bei DKIM erst die gesamte Nachricht übertragen werden muß, bevor die Signatur überprüft werden kann, während bei den beiden anderen im Idealfall bereits der `MAIL FROM`-Befehl ausreicht, die Nachricht zurückzuweisen. Von SPF wird überprüft, ob der SMTP-Client berechtigt ist, eine Nachricht zu versenden, während anhand von DKIM kontrolliert werden kann, ob die Nachricht tatsächlich aus der angegebenen Domain stammt. Aufgrund dieser gegenseitigen Unterstützung kann der parallele Einsatz beider Verfahren nur empfohlen werden, um so die eigene Domäne möglichst vor Adreßmißbrauch zu schützen.

Abschließend soll an dieser Stelle versucht werden, einen kurzen Überblick über die derzeitige Verbreitung der Verfahren DKIM/DomainKeys und SPF/Sender ID zu geben. Leider gibt es hierzu noch keine dedizierten Berichte, weswegen folgendes Zahlenmaterial nicht allzu aussagekräftig ist. In einer Untersuchung[67] von ClickZ zur Zustellbarkeit von E-Mails wird erwähnt,

[66] Siehe [21] S.14f.
[67] Siehe [77].

	nur SPF	nur DKIM	SPF & DKIM	weder noch
Mai 2006	1 487	165	312	35 601
Aug 2006	2 871	195	371	34 127

Tab. 3.8: Domains mit SPF/DKIM-DNS-Records

daß 25% der an der Untersuchung teilgenommenen Firmen wenigstens eines der Verfahren DKIM, SPF oder Sender ID einsetzen. Gemäß der Angaben des TrustedSource-Portals[68] (siehe auch Seite 95) gibt es 42 652 Domänen mit SPF/Sender ID-Einträgen und 2 139 Domänen mit DKIM/DomainKeys-Einträgen. Leider ist keine Angabe zu finden, wieviele Domänen untersucht wurden und von welchem Datum die Zahlen stammen. Da nach meinen Beobachtungen die Angaben innerhalb der letzten sechs Monate nicht aktualisiert wurden, dürften sie aufgrund der Aktivitäten der Arbeitsgruppen und dem entgegenbrachten Interesse inzwischen höher liegen. Im Rahmen der Untersuchungen zu dieser Arbeit (siehe Seite 100) wurden aus den untersuchten Nachrichten alle vorgefundenen Domainnamen extrahiert. Hieraus resultierten 37 565 Domainnamen, für welche am 4.5.2006 und, um Veränderungen zu untersuchen, ein zweites mal am 31.8.2006 getestet wurde, ob SPF- oder DKIM-Informationen per DNS veröffentlicht wurden. Anhand der Ergebnisse (siehe Tabelle 3.8) läßt sich folgendes feststellen:

1. Für beide Verfahren hat die Anzahl der Einträge zugenommen, wobei die Steigerung bei SPF deutlich stärker ausfällt als bei DKIM.

2. SPF weist eine wesentlich stärkere Verbreitung auf.

3. Die Anzahl der Domains, welche wenigstens eines der beiden Verfahren einsetzen, hat sich knapp verdoppelt (von 5,2% auf 9,2%).

4. Fast jede zehnte Domain unterstützt eines der Authentifzierungsverfahren.

Es ist jedoch anzumerken, daß sich die Untersuchung für DKIM etwas schwierig gestaltet, da es möglich ist, daß eine Domain das Verfahren zwar unterstützt, aber keine DNS-Eintrage gefunden werden können, weil hierzu

[68] Siehe [96, 97].

der Wert eines bestimmten Selektors notwendig ist. Es wurden also nur die DKIM-Einträge gefunden, welche direkt unter `_domainkey.domain.tld` veröffentlicht wurden.

Um für die aufgeführten Werte eine Bezugsgröße zu haben, seien folgende Zahlen genannt: Zur Zeit dürfte es weltweit knapp 100 Millionen registrierte Second Level Domains[69] und ca. 440 Millionen Hostnamen[70] geben. Damit ist die Verbreitung von SPF/Sender ID bzw. DKIM nach wie vor noch sehr gering, was sich jedoch aufgrund der beobachteten Wachstumsraten schnell ändern könnte.

[69] Siehe [17]. Die zehn größten Top Level Domains haben zusammen über 91 Millionen registrierte Domänen.

[70] Siehe [47].

Reputation, reputation, reputation! Oh, I have lost my reputation! I have lost the immortal part of myself, and what remains is bestial.

Cassio, in Shakespeares Othello

4 Reputationsverfahren

Wie im vorangegangenen Kapitel dargelegt wurde, leistet die Authentifizierung des Absenders einen wertvollen Beitrag bei der Lösung des Spamproblems. Anhand der vorgestellten Verfahren kann sicherlich erreicht werden, daß die Gefahr durch Würmer und Trojaner verringert wird, denn kaum ein Autor von Malware wird diese per Mail verbreiten, wenn die Herkunft ohne große Schwierigkeiten feststellbar ist. Es gibt aber genügend Firmen und Spamversender, welche ganz offen per E-Mail werben und den Ursprung nicht verschleiern bzw. es ist davon auszugehen, daß auch Spammer die Authentifizierungsmerkmale einsetzen, um so möglicherweise eine bessere Zustellrate für ihre Werbemails zu erreichen.

Tatsächlich ist es bei Spammern bereits eine verbreitete Strategie, Domains nur für einen kurzen Zeitraum von wenigen Stunden zu benutzen und von diesen um so aggressivere Spamwellen zu lancieren. Diese Spamdomains werden meist nicht länger als zwölf Tage tatsächlich genutzt[71]. Bei den im

[71] Siehe [60] S.7.

Rahmen der Arbeit untersuchten Spam-Mails fand sich zwar nur ein geringer Anteil an Mails mit erkennbaren Authentifizierungsmerkmalen (DKIM und SPF), aber mit zunehmender Verbreitung dieser beiden Verfahren dürfte auch Spam diese Kennzeichen aufweisen. Falls die Versender von Spam in Zukunft den Aufwand scheuen sollten, für ihre kurzlebigen Spamdomains entsprechende Schlüssel und DNS-Einträge zu erstellen, könnte so ein erster Erfolg im Kampf gegen Spam erzielt werden.

Festzuhalten bleibt jedoch, daß eine erfolgreiche Authentifikation des Absenders nicht automatisch bedeutet, daß die überprüfte Mail keine unerwünschte Nachricht ist. Gerade in diesem Fall werden weitere Hilfsmittel benötigt, anhand deren die E-Mail klassifiziert werden kann. Hierfür können Reputationssysteme einen wertvollen Beitrag leisten.

4.1 Existierende Reputationssysteme

Auch wenn die meisten Spammer versuchen, die Herkunft ihrer Nachrichten zu verschleiern, läßt sich über die in den Received-Headern einer E-Mail gespeicherten IP-Adressen der Weg bis zu einem gewissen Punkt verläßlich zurückverfolgen. Auf diese Weise erhält man letztlich zumindest einen Provider, welcher für die Einspeisung der E-Mail verantwortlich ist.

Diese Tatsache und weitere Ergebnisse, welche aus der Analyse größerer Nachrichtenmengen erzielt werden, werden bereits genutzt und kommen bei RBLs und kostenpflichtigen Reputationsdiensten zum Einsatz.

4.1.1 Realtime Blackhole Lists

Eine Realtime Blackhole List (RBL) – oder auch DNS-based Blackhole List (DNSBL) – stellt im Grunde eine einfache Reputationsdatenbank dar. Die erste RBL wurde 1997 von Paul Vixie ins Leben gerufen, mit dem Ziel, Internet Provider auf offene SMTP-Server innerhalb ihres Administrationsbereiches aufmerksam zu machen. Inzwischen gibt es zahlreiche Betreiber solcher Listen, die bekanntesten sind SORBS [71], Spamcop [72] und The Spamhaus Project [92]; einen ausführlichen Überblick über weitere RBLs geben [15]

und [18].

Die Funktionsweise ist bei allen Listen ähnlich: In einer Liste werden jene IP-Adressen gespeichert, die durch massive Spamverbreitung auffallen. Während einer SMTP-Sitzung kann der empfangende Mailserver überprüfen, ob die IP-Adresse des Absenders auf einer (oder auch mehreren) RBL gelistet ist, und je nach Ergebnis die Annahme verweigern oder – besonders problematisch – stillschweigend die Nachricht löschen.

Die meisten RBL-Betreiber veröffentlichen ihre IP-Listen per DNS-Server. Demnach wird die Information, ob eine bestimmte IP-Adresse auf einer RBL eingetragen ist, meist über eine DNS-Anfrage erhalten. Soll beispielsweise die IP-Adresse 137.248.121.158 aus der Domain mathematik.uni-marburg.de überprüft werden, wird an den Server des Listenbetreibers eine Anfrage nach einem A-Record für `158.121.248.137.mathematik.uni-marburg.de` gestellt. Die Antwort enthält entweder „No such domain“, d. h. die IP-Adresse ist nicht gelistet, oder eine IP-Adresse, d. h. der entsprechende Host ist in der RBL gespeichert. Die zurückgelieferte IP-Adresse stammt aus dem Loopback-Netzwerk 127.0.0.0/8. Damit werden – ähnlich wie HTTP-Statuscodes – genauere Informationen über den Host übermittelt.

Die RBLs unterscheiden sich stark anhand der Richtlinien, wann eine Adresse in die Datenbank aufgenommen wird und wann und wie sie wieder daraus gelöscht wird. Man kann nach folgenden Gesichtspunkten differenzieren:

1. Welche Hosts werden gelistet? (Open Relay, Proxy, Host eines Spammers etc.)

2. Wie werden die IP-Adressen gesammelt? (automatisch, von Usern gemeldet, Honeypot)

3. Wie lange verbleibt die Adresse in der Liste? (automatische versus manuelle Entfernung)

4. Was kann ein Administrator einer gelisteten IP-Adresse tun, um wieder von der Liste gelöscht zu werden?

Das größte Problem der RBL sind fälschlich gelistete IP-Adressen [28,29]. Sind hiervon große Mailprovider wie z. B. Yahoo! oder GMX betroffen, kann es vorkommen, daß der Absender der E-Mail von deren Abweisung nichts merkt. Der Empfänger wird nichts bemerken, weil er in der Regel keinen Einblick in die Logdateien nehmen kann und weil eine Systemmeldung über eine verweigerte Annahme letztendlich die gleiche Störung verursacht wie die abgewiesene Nachricht. Durch das Blockieren von Nachrichten sinkt letztlich das Vertrauen in die Verläßlichkeit dieses Mediums. Aus diesem Grund wird empfohlen, eine RBL nicht zum Abweisen von Nachrichten zu nutzen, sondern lediglich im Rahmen eines Bayesischen Filters um den Spamscore anzupassen.

RBLs bieten insgesamt eine recht gute Möglichkeit, Spam gar nicht erst beim Empfänger ankommen zu lassen, die Erfolgsquote liegt bei ca. 80% [9]. Allerdings sind sie – wenn sie zur Löschung von Mails herangezogen werden – in ihrem Einsatz nicht unproblematisch. Und wenn sie nur zur Veränderung eines Spamscores verwendet werden, eignen sie sich lediglich für den Einsatz an zentraler Stelle auf dem Mailserver eines Providers. Denn personalisierte, also auf den Mailverkehr eines Benutzers trainierte Spamfilter erreichen eine Erfolgsquote von über 98%. Darüber hinaus sind RBL relativ langsam und können lediglich reagieren, d. h. erst wenn Spam-Mails in der Inbox gelandet sind und der Absender dem RBL-Betreiber gemeldet wurde, können sie ihre Wirkung entfalten.

4.1.2 Reputation Service Provider

Alle großen E-Mail-Provider haben inzwischen ein Anti-Spam-Modul im Angebot, welches von den Kunden genutzt werden kann, doch diese Provider richten ihre Angebote meist nur an Privatkunden. Für Firmen stellt sich das Problem, daß sie selbst eine Infrastruktur zur Spamverhinderung aufbauen müßten. Dem Problem, daß manche E-Mail den Adressaten nicht erreicht, bzw. daß nicht jede E-Mail angenommen werden soll, haben sich inzwischen mehrere Firmen angenommen. Mittlerweile gibt es verschiedene Angebote für Firmen rund um den Versand und Empfang von E-Mails.

4.1.2.1 Dienstleistungen für Empfänger

Firmen wie TrendMicro [61], IronPort [48] und CipherTrust [11] bieten in diesem Zusammenhang Dienste an, welche versprechen, unerwünschte E-Mails von den Postfächern ihrer Kunden weitgehend fernzuhalten. Durch eine weltweite Analyse des Mailaufkommens (Datenverkehr, Abgleich mit Black- und Whitelists, Netzwerkcharakteristik, Senderverhalten, Mailcharakteristik) an mehreren hundert oder tausend Punkten[72] ihres Netzwerks werden die IP-Adressen der Absender klassifiziert. Das Ergebnis ist ein Reputationssystem, anhand dessen eingehende Mail analysiert werden kann und so genauere Aussagen ermöglicht, als eine einfache RBL[73].

4.1.2.2 Dienstleistungen für Absender

Es gibt aber auch Anbieter, welche sich stärker an den Bedürfnissen von Absendern orientieren. Die Produkte von Unternehmen wie ReturnPath [83], Goodmail Systems [24] oder Habeas [27] sind auf Firmen ausgerichtet, welche vermeiden wollen, daß ihre Nachrichten und Werbesendungen in Spamfiltern hängen bleiben. Vor allem die von Habeas angebotenen Dienste sind entsprechend vielfältig. Von der Zertifizierung des Kunden (Analyse der Reputation, Einhaltung von Rechtsnormen und Standards, Whitelisting), Hilfe bei Zustellungsproblemen (in Zusammenarbeit mit großen ISPs) über die Auswertung von E-Mail-Werbekampagnen bis hin zu einer Vorab-Analyse der zu versendenden Nachricht wird nahezu alles unternommen, um den Prozentsatz der nicht zugestellten E-Mails möglichst gering zu halten.

In einem ähnlichen Zusammenhang ist auch das Positivlistenprojekt Certified Senders Alliance zu sehen. Dieses Bündnis zwischen Direktvermarktern und Providern wurde auf dem zweiten deutschen Anti-Spam-Kongress im September 2004 gegründet. Durch Double-Opt-in, klarer Kennzeichnung

[72] CipherTrust hat nach eigenen Angaben in 40 Ländern über 4000 Sensoren in 1600 Firmen installiert. Ein Drittel dieser Firmen stammt aus den Fortune 500.

[73] Einen kleinen Einblick in die von CipherTrust gewonnen Daten erhält man auf der Portalseite [98] ihrer „TrustedSource“ genannten Anti-Spam-Technologie. Dort kann zu einem Domainnamen die Abweichung vom durchschnittlichen Mailaufkommen, die Anzahl der sendenden IP-Adressen etc. abgefragt werden. Zu einer IP-Adresse erhält man zusätzlich die Reputation.

und einem Mitgliedsbeitrag „erkaufen“ sich die Marketingfirmen die Möglichkeit, daß ihre Werbesendungen nicht in den Filtern der beteiligten Provider hängen bleiben; lediglich individuelle Filter der Nutzer dürfen diese Nachrichten dann noch behindern.

4.1.2.3 Diskussion

Jene Reputation Service Provider (RSP), welche die Zustellungsquote der E-Mails ihrer Kunden verbessern möchten, leisten für den Empfänger der Nachricht einen eher zweifelhaften Nutzen. Von Seiten der Dienstleister wird stets betont, daß die Zahl der False Positives reduziert würde, was letztendlich auch dem Empfänger zugute käme. Für die meisten Anwender dürfte die Anzahl der Firmen eher gering sein, deren erwünschte Werbemails wahrscheinlich in einem Spamfilter hängen bleiben. Dieses Problem ließe sich über eine individuelle Whitelist einfacher und vor allem billiger lösen.

Die „guten“ Reputation Service Provider leisten brauchbare Dienste dabei, jene Spam-Mails zu verhindern, bei denen versucht wird, die Herkunft zu verschleiern (Zombie-PCs, Open Relays etc.). Inwiefern sie ihre Wirksamkeit gegen Werbemails von Direktmarketingfirmen zum Einsatz bringen können und dürfen, hängt von den jeweiligen rechtlichen Gegebenheiten bzw. von den Kunden dieser Firmen ab.

Ferner ist es notwendig, die Rolle eines RSP genauer zu betrachten. Erstens ist festzuhalten, daß dieser kostenpflichtige Dienstleistungen erbringt und somit nur von Firmen und größeren Organisationen beauftragt werden kann, kaum aber von Privatanwendern. Zweitens muß die Frage gestellt werden, wie weit einem RSP vertraut werden kann. Zum einen könnte sich eine Firma eine gute Reputation bei diesem RSP „erkauft“ haben, damit deren E-Mails ungefiltert passieren können, zum anderen wird durch den RSP möglicherweise geschäftssensibler Mailverkehr analysiert, was das Ausspähen von Firmengeheimnissen und/oder -kontakten ermöglicht. Drittens und letztens besteht, wenn sich das Geschäftsmodell eines RSP durchsetzt, die Gefahr, daß ein Großteil des weltweiten Mailverkehrs durch wenige Reputationsdienstleister analysiert und klassifiziert werden könnte. Einerseits entstünde dadurch

ein großes Machtvolumen, welches Mißtrauen hervorrufen würde, andererseits wäre ein Reputationssystem, welches von wenigen zentralen Instanzen kontrolliert wird, anfälliger gegenüber Störaktionen und Ausfällen als ein engmaschiges Netz zahlreicher unabhängiger Akkreditierungsdienste.

4.2 Vorschlag für ein verteiltes Reputationsverfahren

Fast jede E-Mail durchläuft einen Bayesischen Filter, bevor die Nachricht beim Empfänger eintrifft. Dies geschieht entweder auf dem lokalen Rechner des Benutzers oder auf einem Mailserver der Firma bzw. des ISP. Abgesehen von einem Eintrag in den Headerzeilen und der dadurch möglichen Filterung oder Löschung der Nachricht, bleiben die Ergebnisse dieses Vorgangs ansonsten ungenutzt. Ziel des hier erläuterten Vorschlags ist es, die durch den Bayesischen Filter gewonnenen Informationen zu nutzen und in Verbindung mit den Eigenschaften sozialer Netze ein „Web of Trust“ aufzubauen, in dem jeder seinem nächsten Bekannten vertraut, mit dem er per E-Mail kommuniziert. Dieses ermöglicht Rückmeldungen von den Empfängern und eine verteilte Reputationsdatenbank mit der Möglichkeit für Reputationsauskünfte.

4.2.1 Überblick über das Verfahren

Um dem Absender einer Nachricht eine gute oder schlechte Reputation zusprechen zu können, ist es wichtig, daß dieser zweifelsfrei festgestellt werden kann. Dieser Vorschlag geht davon aus, daß Authentifizierungsmechanismen wie die in Kapitel 3 vorgestellten sich durchsetzen und verstärkt zum Einsatz kommen werden.

Für eine eintreffende E-Mail (vgl. Abbildung 4.1, Seite 98) wird zunächst mittels eines Bayesischen Filters bestimmt, mit welcher Wahrscheinlichkeit diese Nachricht als Spam einzuordnen ist. Dieser Wert wird im folgenden als Spamwert (SW) bezeichnet. Anschließend wird geprüft, ob der Absender authentifizierbar ist. Der Fall, daß keine Authentifizierungsinformationen

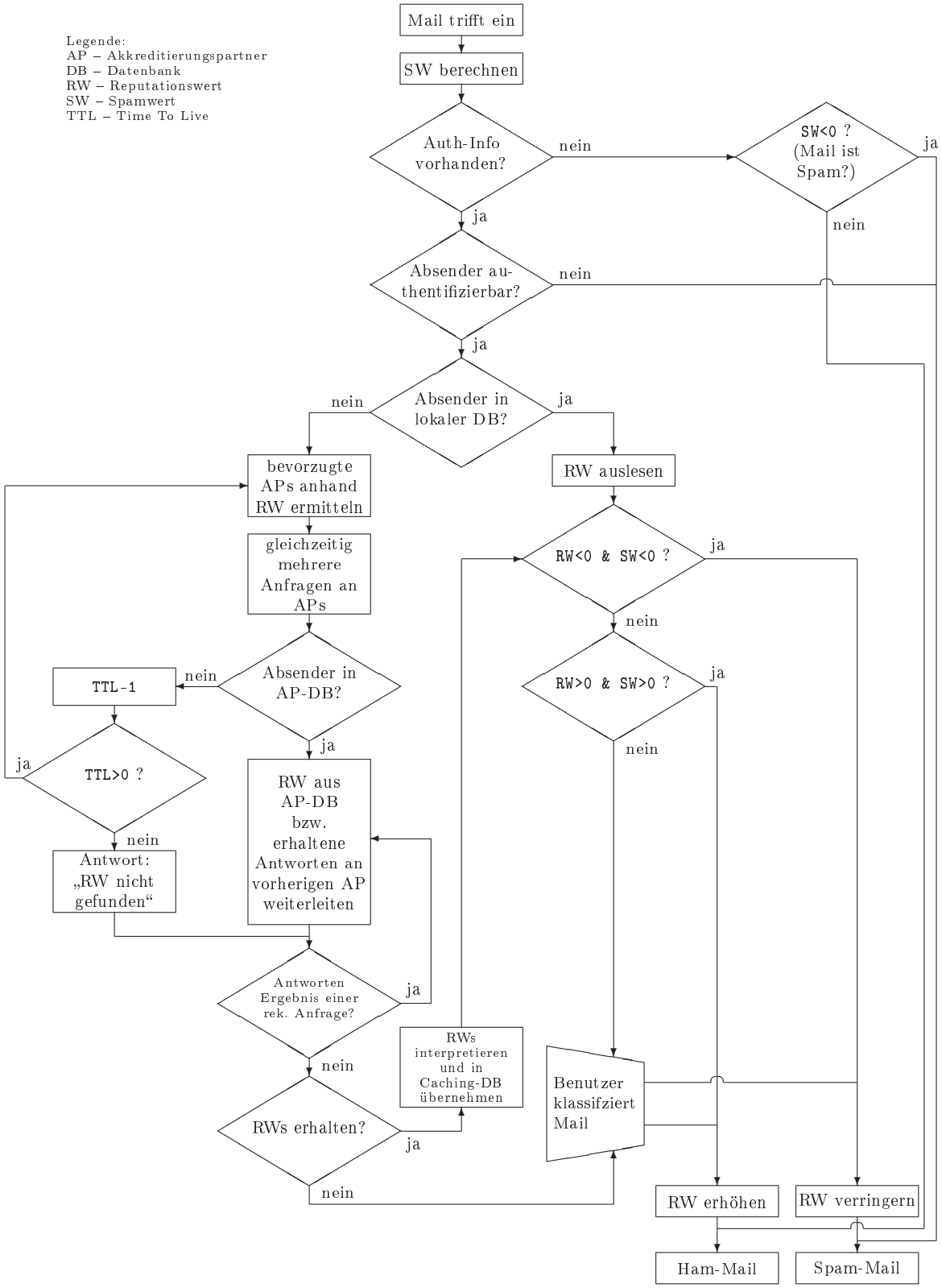

Abb. 4.1: Ablaufdiagramm

vorliegen, wird im Abschnitt 4.2.5 diskutiert. Schlägt die Authentifizierung fehl, ist dies ein starkes Indiz für gefälschte Absenderinformationen, die Nachricht sollte als unerwünscht eingestuft werden, vor allem, wenn der Spamwert ebenfalls auf dieses Ergebnis hinweist. Im Zweifelsfall kann die endgültige Entscheidung dem Benutzer überlassen werden.

Sollte die Authentifizierung jedoch erfolgreich sein, wird in einer lokalen Datenbank die Reputation des Absenders nachgeschlagen. Weisen der Reputationswert (RW) und der Spamwert beide auf das gleiche Ergebnis hin (Spam bzw. Ham), wird die Nachricht entsprechend behandelt, d. h. Spam wird in einen speziellen Ordner verschoben oder gelöscht und Ham wird normal der Inbox zugestellt und der RW des Absenders je nach Ergebnis erhöht oder erniedrigt. Sollten sich RW und SW widersprechen, ist eine Klassifizierung durch den Benutzer notwendig, um entweder den Spamfilter zu trainieren oder fehlerhafte Auswirkungen auf den Reputationswert zu verhindern. Es ist aber auch eine differenziertere Vorgehensweise denkbar, bei der RW und SW genauer analysiert werden. Weist z. B. der SW nur schwach auf eine unerwünschte E-Mail hin, der RW hingegen drückt eine ausgesprochen positive Reputation aus, so könnte die Nachricht auch ohne Interaktion des Benutzers als Ham gewertet werden.

Bis zu diesem Punkt entspricht dieses Verfahren einer lokalen, automatisierten Black-/Whitelist. Über authentifizierte, aber unbekannte Absender kann keine Aussage getroffen werden, da in der lokalen Datenbank keine Reputationswerte gespeichert sind. Um dieses Problem zu lösen, wird versucht, durch rekursive Akkreditierungsanfragen eine Aussage über die Reputation des Absenders zu erhalten. Wichtige Voraussetzung hierfür ist, daß eine Reputationsdatenbank auch Anfragen Dritter zur Verfügung steht und somit als Akkreditierungsdienst eingesetzt werden kann.

Um diesen Akkreditierungsschritt durchzuführen, wird eine Anfrage an einen oder mehrere bevorzugte Akkreditierungspartner (AP) geschickt. Findet sich die Mailadresse in der Datenbank des AP, wird deren RW als Antwort verschickt. Kann kein RW gefunden werden, stellt der AP seinerseits Anfragen an die von ihm bevorzugten Akkreditierungspartner, welche wiederum genauso vorgehen wie der erste AP. Sobald ein AP eine Antwort erhält, lei-

	Mails gesamt	Mails mit extrahierbaren Adressen	verschiedene Adressen	verschiedene Domains
Ham	39 099	38 230	5 666	3 184
Spam	31 579	31 417	29 951	14 161

Tab. 4.1: Analyse der From-Adresse

tet er diese an die anfragende Stelle weiter. Eine genauere Darstellung dieses Lookupmechanismus findet sich im Abschnitt 4.2.4.

Am Ende des Abfragevorgangs treffen am Ausgangspunkt eine oder mehrere Antworten ein. Konnte keine Anfrage einen Reputationswert liefern oder sind die Antworten zu wenig aussagekräftig oder widersprüchlich, könnte nur der Spamwert einen Indiz liefern. Da über diesen bislang unbekannten Absender im lokalen Datenbestand keine Informationen vorliegen und nun erstmalig eine Reputation bestimmt werden soll, sollte die Klassifikation auf jeden Fall durch den Benutzer kontrolliert werden. Falls die Anfrage jedoch eindeutige Ergebnisse geliefert hat, wird die Nachricht dementsprechend als Ham oder Spam behandelt und die entsprechenden Informationen im lokalen Datenbestand gespeichert.

Bevor einzelne Begriffe und Vorgehensweisen genauer erläutert werden, wird im nächsten Abschnitt auf das unterschiedliche Sendeverhalten von Spam- und Ham-Versendern eingegangen.

4.2.2 Analyse des Sendeverhaltens

Folgende Zahlen stammen aus der Analyse des Mailaufkommens verschiedener E-Mailadressen des Autors. Das Mailaufkommen besteht deshalb aus einer Mischung zahlreicher Mailarten: Spam-Mails, Mails von teilweise sehr stark frequentierten Mailinglisten, Mails von privaten Adressen mit hohem und niedrigem Aufkommen, geschäftliche Nachrichten, teils von regelmäßigen, teils von singulären Kontakten. Insgesamt wurden 70 678 Nachrichten analysiert, davon 31 579 Spam-Mails und 39 099 Ham-Mails.

Auch wenn zur Zeit nur sehr wenige E-Mails mit Authentifizierungsmerkmalen versehen sind und somit keine sicheren Aussagen über den tatsächlichen Absender möglich sind, liefert die Analyse der From-Adresse signifikan-

te Unterschiede zwischen Ham und Spam. In Tabelle 4.1 ist, nach Ham und Spam getrennt, die Häufigkeit unterschiedlicher Mailadressen und Domainnamen dargestellt. Die dritte Spalte gibt dabei an, aus wie vielen E-Mails eine From-Adresse extrahiert werden konnte. Der Unterschied zur jeweiligen Gesamtzahl erklärt sich durch fehlende oder nicht-RFC-konforme Adressen, wie z. B. „`tommie at domain.com`“ oder „`‘‘Olaf.Schmidt.‘‘@domain.de`“, mit denen der Absender versucht, seine Adresse vor Suchmaschinen zu verbergen.

Auffällig ist, daß die Werte für die Anzahl der unterschiedlichen Absenderadressen und der Domainnamen (Spalten vier und fünf) bei Spam-Mails deutlich höher als bei Ham-Mails ausfallen. Dies legt die Vermutung nahe, daß Spamversender eher selten die gleiche Adresse mehrmals benutzen, während man von einem Hamversender meist mehrere Mails bekommt.

Mail-aufkommen	Anzahl Adressen Ham	Anzahl Adressen Spam	Anteil an Gesamtanzahl Ham	Anteil an Gesamtanzahl Spam
1	2880	29 216	50,8295%	97,5460%
2	875	669	15,4430%	2,2336%
3	406	22	7,1655%	0,0735%
4	284	21	5,0124%	0,0701%
5	200	1	3,5298%	0,0033%
6	145	2	2,5591%	0,0067%
7	113	1	1,9944%	0,0033%
8	83	1	1,4649%	0,0033%
9	62	0	1,0942%	0,0000%
10	68	2	1,2001%	0,0067%
11-20	269	8	4,7476%	0,0267%
21-50	148	6	2,6121%	0,0200%
51-100	79	0	1,3943%	0,0000%
über 101	54	2	0,9531%	0,0067%

Tab. 4.2: Anzahl unterschiedlicher Absenderadressen, kategorisiert nach Anzahl eingegangener Mails

Zur Überprüfung dieser Aussage wurden die Absenderadressen analysiert und gezählt, wieviele Mails von jeder Adresse und jeder Domain verschickt wurden. In Tabelle 4.2 ist aufgeschlüsselt, von wie vielen verschiedenen Absenderadressen eine, zwei, drei, etc. Mails empfangen wurden. Beispielsweise

Mail-auf-kommen	Anzahl Domains Ham	Anzahl Domains Spam	Anteil an Gesamtanzahl Ham	Anteil an Gesamtanzahl Spam
1	1 483	10 824	46,5766%	76,4353%
2	484	1 673	15,2010%	11,8141%
3	243	605	7,6319%	4,2723%
4	163	327	5,1193%	2,3092%
5	107	176	3,3606%	1,2429%
6	94	98	2,9523%	0,6920%
7	78	79	2,4497%	0,5579%
8	56	55	1,7588%	0,3884%
9	42	54	1,3191%	0,3813%
10	38	39	1,1935%	0,2754%
11-20	171	116	5,3706%	0,8192%
21-50	111	73	3,4862%	0,5155%
51-100	57	29	1,7902%	0,2048%
über 101	57	13	1,7902%	0,0918%

Tab. 4.3: Anzahl unterschiedlicher Absenderdomains, kategorisiert nach Anzahl eingegangener Mails

wurde von 2 880 verschiedenen Absenderadressen jeweils nur eine Ham-Mail empfangen, aber von 29 216 unterschiedlichen Adressen genau eine Spam-Mail. Die Spalten vier und fünf zeigen jeweils den prozentualen Anteil an der Zahl der Ham- bzw. Spam-Adressen, um der Tatsache gerecht zu werden, daß die Anzahl der Spam-Mails geringer ist als die Anzahl der Ham-Mails, womit sich natürlich auch ein quantitativer Unterschied bei den Absenderadressen ergibt. Tabelle 4.3 zeigt die korrespondierenden Zahlen für die Absender*domains*. Da die Anzahl der Absender mit mehr als zehn versandten Nachrichten eher gering ist, wurden die entsprechenden Häufigkeiten aufsummiert. Die vollständigen, ungekürzten Tabellen finden sich im Anhang (siehe Seite 133 bzw. Seite 137).

Aus den Zahlen wird ersichtlich, daß die meisten Spam-Mails unterschiedliche Absenderadressen und -domains aufweisen, Spammer also eher selten eine Adresse mehrmals benutzen (weniger als 3%). Für Ham-Mails gilt im Prinzip die umgekehrte Aussage, allerdings ist der Anteil der Absender, von denen man nur einmal eine Ham-Mail erhält, mit über 50% noch relativ

hoch. Als Faustregel kann jedoch festgehalten werden: Spam-Mails erhält man von vielen unbekannten Absendern, Ham-Mails hingegen eher von bereits bekannten Absendern. Die Häufigkeit, wie oft man von einer bestimmten Absenderadresse E-Mails erhält, kann also ein Indiz sein, ob eine Nachricht als Spam oder Ham zu werten ist.

Dies nutzen manche Mailempfänger für eine – allerdings nicht ganz unproblematische – Anti-Spam-Maßnahme: Mails von unbekannten Absendern werden ignoriert und erst beim zweiten oder dritten Kommunikationsversuch wird auf die E-Mail geantwortet. Der Fehler, der mit dieser Methode gemacht

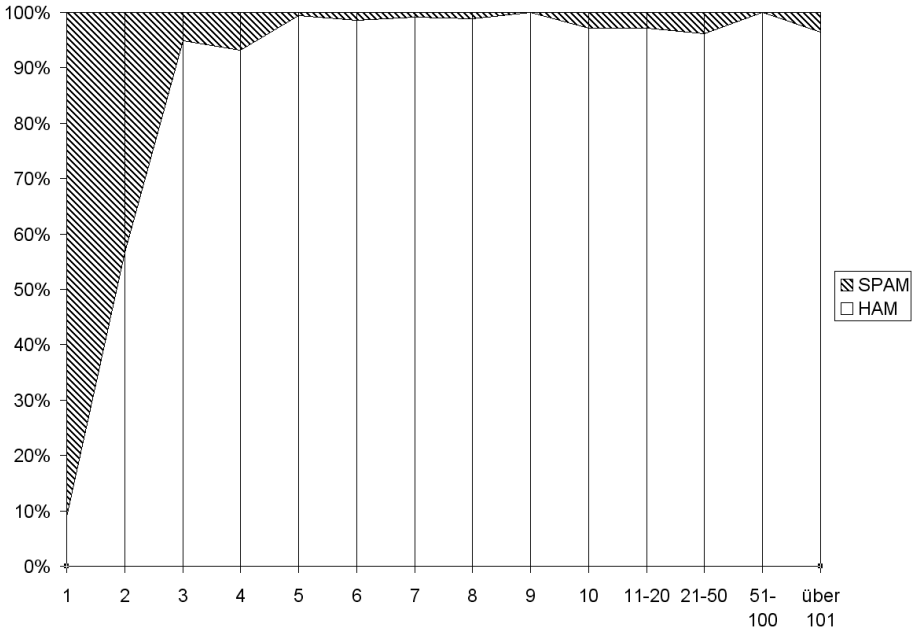

Abb. 4.2: Absender von Spam- versus Ham-Mails, nach Mailaufkommen kategorisiert

wird, ist geringer als zunächst zu erwarten wäre. Wie Abbildung 4.2 zeigt, ist die Wahrscheinlichkeit, daß eine erhaltene E-Mail Ham ist, wenn die Absenderadresse unbekannt ist, kleiner als 10% (genauer Wert 8,97%). Ist die erhaltene Mail die zweite Nachricht von dieser Adresse, steig die Wahrscheinlichkeit auf 56,67%, ist es die dritte Nachricht auf 94,85%.

Diese Zahlen werden sich mit dem zunehmenden Einsatz von Autorisierungsverfahren sicherlich ändern, da dann der Zugang zu „Einmaladressen“ für Spammer schwieriger und letztendlich nur noch über Spamdomains (sie-

he Seite 91) möglich sein wird. Damit bleiben zwei Möglichkeiten: Entweder Spamdomains verbreiten sich zusehends, dann würde sich an obigen Zahlen wenig ändern, oder Spammer verwenden öfters die gleichen Absenderadressen, wodurch bessere Reputationsaussagen möglich werden.

Eine interessante Beobachtung ergibt sich bei einer genaueren Betrachtung der 13 Domains, von denen jeweils mehr als 100 Spam-Mails empfangen wurden (siehe Tabelle 4.4). Unter diesen Domainnamen sind vor allem große

Ham	Spam	Domain
516	1 422	yahoo.com
537	1 246	hotmail.com
42	342	msn.com
884	331	gmx.net
2	287	netscape.net
0	258	prodigy.net
0	227	ientrynetwork.net
7	178	verizon.net
3464	175	gmail.com
819	113	mathematik.uni-marburg.de
98	113	aol.com
26	111	excite.com
103	107	comcast.net

Tab. 4.4: Domains mit höchstem Spamaufkommen

E-Mailprovider vertreten: Yahoo!, Hotmail, MSN, GMX, Netscape, Google-Mail und AOL. Dies läßt sich dadurch erklären, daß Spammer gerne Adressen dieser Domains als Absender verwenden bzw. im From-Header eintragen, wobei es egal ist, ob die Adresse tatsächlich existiert oder frei erfunden ist. Das sehr hohe Hamaufkommen der Domain `gmail.com` liegt speziell in diesem Fall an sehr vielen E-Mails, welche über stark frequentierte Mailinglisten von Nutzern mit GMail-Adresse empfangen wurden. Ähnliches gilt für GMX. An diesen beiden Punkten wird deutlich, daß die Daten auf dem Mailaufkommen eines einzelnen Empfängers basieren. Hierzu wären Untersuchungen direkt an einem Mailserver interessant, um einen domänenweiten Überblick zu erhalten.

Des weiteren fällt die Domain `mathematik.uni-marburg.de` auf, aus der

die Adresse des Autors stammt und für die mit 113 Mails ebenfalls ein recht hohes Spamaufkommen zu verzeichnen ist. Da Spammer gerne Absenderadressen vorgeben oder fälschen, welche aus derselben Domain stammen wie die Empfängeradresse, oder als Absender die Adresse des Empfängers eintragen, kann mit diesen beiden Fälschungsmöglichkeiten das hohe Spamaufkommen aus dieser Domain erklärt werden.

Eine weitere These, worin sich Spam von Ham unterscheiden könnte, betrifft die Anzahl der Adressaten: Spam-Versender verschicken *eine* Nachricht an *viele* Adressen, während Ham-Versender *mehrere* Nachrichten an *wenige* Adressen schicken.

Anzahl Adressaten	Anteil bei Ham	Anteil bei Spam
1	61,27%	80,45%
2	19,89%	5,69%
3	8,31%	1,84%
4	5,62%	1,43%
5	1,59%	1,69%
6	0,53%	1,44%
7	0,20%	1,64%
8	0,17%	1,25%
9	0,08%	1,25%
10	0,07%	0,61%
11	0,07%	0,51%
12	0,05%	0,37%
13	0,02%	0,17%
14	0,01%	0,14%
15	0,01%	0,11%

Tab. 4.5: Prozentsatz der Nachrichten mit einem, zwei, drei etc. Adressaten

Diese Vermutung ließ sich anhand der zur Verfügung stehenden Daten jedoch nicht belegen, wie Tabelle 4.5 zeigt. Dargestellt ist der Anteil der Ham- bzw. Spam-Mails mit einem, zwei, drei etc. Adressaten (extrahiert aus To- und CC-Header). Demnach enthält eine Spam-Mail im Durchschnitt 1,7889 Empfänger und eine Ham-Mail 1,7903 Empfänger.

Es muß allerdings angemerkt werden, daß eine exakte Analyse nur aufgrund des To-und CC-Headers nicht möglich ist. Einerseits können weitere

Adressaten im BCC-Header genannt sein, welcher beim Empfänger nicht ersichtlich ist. Andererseits besteht die Möglichkeit, daß Nachrichten mit identischem Inhalt in anderen SMTP-Sitzungen verschickt wurden. Auch hier hilft nur die Analyse von Daten, welche direkt am Mailserver oder einem Mailrelay gesammelt werden.

4.2.3 Grundlagen und Begriffe

In den folgenden Abschnitten werden zentrale Bestandteile und Begriffe des Reputationsverfahrens erläutert und genauer gefaßt.

4.2.3.1 Absender und Reputationsidentität

Für die gewünschten Reputationsaussagen ist es wichtig, den Absender einer E-Mail korrekt zu erfassen. Hierzu leisten die Verfahren SPF/Sender ID und DKIM einen entscheidenden Beitrag, sofern sich der Absender gegenüber der überprüften Domain authentifiziert hat. Aus diesem Grund wird an dieser Stelle gefordert, daß für jede Domäne, welche Authentifizierung per DKIM oder SPF/Sender ID ermöglicht, zwingend eines der auf Seite 87 erwähnten Verfahren eingesetzt wird.

Als Absender wird stets jene Adresse herangezogen, welche durch das jeweils zum Einsatz kommende Verfahren authentifiziert werden konnte. Das bedeutet, bei SPF/Sender ID wird die `MAIL FROM`- bzw. PRA-Adresse verwendet, bei DKIM kann, je nach Inhalt des i-Tags die `From`- oder `Sender`-Adresse herangezogen werden.

Von dieser Adresse wird zum einen die Mailadresse selbst und zum anderen der Domainname verwendet. Dadurch sind zwei unterschiedliche Anfragen und Reputationsaussagen möglich: Aussagen über die Domain und, falls zur genaueren Differenzierung nötig, Aussagen über die Adresse[74].

Diese Unterscheidung ist wichtig, wie z. B. die Security Bulletins von Microsoft zeigen. Bis zum März 2006 verwendete Microsoft für jeden Newsletter eine Adresse der Form `<localpart>@newsletters.microsoft.com`, wobei

[74] Falls keine Mailadresse zur Verfügung steht (leerer `MAIL FROM`-Befehl bei SPF/Sender ID), kann eine Reputationsaussage nur bezüglich der Domain getroffen werden.

<localpart> stets geändert wurde. Ohne Reputationsaussagen zur Domain wäre die Klassifizierung einer solchen Nachricht deutlich unsicherer.

4.2.3.2 Strukturierung und Elemente der Datenbanken

Zur Unterscheidung eigener und fremder Daten werden zwei Datenbanken verwendet, eine Primärdatenbank, in der die Reputationsinformationen aufgrund eingegangener E-Mails gespeichert werden, und eine Datenbank, in der die Ergebnisse aus Akkreditierungsanfragen gespeichert werden, hier Caching-Datenbank genannt. Die Datenbanken enthalten im wesentlichen folgende Elemente:
Elemente der Primärdatenbank:

- **Mailadresse** des authentifizierten Absenders
- Aus der Mailadresse extrahierter **Domainname**
- **Adresse des Akkreditierungsdienstes**: (IP-)Adresse, an die Akkreditierungsanfragen gestellt werden können
- Ein Zähler für die **Anzahl Ham**-Mails
- Ein Zähler für die **Anzahl Spam**-Mails. „Anzahl Ham“ und „Anzahl Spam“ dienen zur Berechnung des Reputationswertes (siehe 4.2.3.3).

Elemente der Caching-Datenbank:

- **Mailadresse** des authentifizierten Absenders
- Aus der Mailadresse extrahierter **Domainname**
- **Reputationswert Adresse** Durch eine Reputationsauskunft erhaltener Wert für die Reputation dieser Adresse
- **Reputationswert Domain** Durch eine Reputationsauskunft erhaltener Wert für die Reputation dieser Domain

Die Caching-Datenbank dient lediglich dazu, Akkreditierungsanfragen zu beantworten, zu denen zwar keine Ergebnisse in der Primärdatenbank vorliegen, aber Antworten aus früher gestellten Anfragen. Für die Klassifizierung einer eingegangenen Nachricht wird zunächst in der Primärdatenbank nachgeschlagen. Liegen dort keine Informationen vor, wird nicht auf eventuell in der Caching-Datenbank vorhandene Werte zurückgegriffen, sondern eine Akkreditierungsanfrage gestellt (siehe Kapitel 4.2.4). Dies hat den Hintergrund, daß die Werte in der Caching-Datenbank veraltet sein könnten. Wenn die Nachricht klassifiziert wurde und in der Caching-Datenbank Informationen zu dem Absender der Nachricht vorliegen, werden diese gelöscht und ein neuer Eintrag in der Primärdatenbank angelegt, was einen reinigenden Effekt für den Cache hat.

4.2.3.3 Der Reputationswert

Der Reputationswert (RW) ist eine Zahl, welche für Absenderadressen (bzw. Domainnamen) berechnet oder in einer Akkreditierungsantwort mitgeteilt werden kann, und drückt die Reputation der Adresse bzw. der Domain aus. Die Mitteilung der absoluten Anzahl der Spam- und Ham-Mails in einer Antwort auf eine Akkreditierungsanfrage würde möglicherweise auf Widerstand bei sicherheitsbewußten Naturen stoßen, die einfache Differenz hingegen hat zu wenig Aussagekraft, weil nicht ersichtlich ist, wie groß der Anteil an Spam- bzw. Ham-Mails ist. Aus diesem Grund wird die Differenz zum Gesamtaufkommen ins Verhältnis gesetzt und zusätzlich aus dem resultierenden Wert die Wurzel gezogen:

$$rw(h,s) = \begin{cases} \sqrt{\frac{h-s}{h+s}} & \text{, falls } h \geq s \\ -\sqrt{\frac{s-h}{h+s}} & \text{, falls } h < s \end{cases}$$

Dabei bezeichnet h die Anzahl der Ham-Mails und s die der Spam-Mails (siehe auch Abbildung 4.3). Falls h und s beide Null sind, ist der Reputationswert nicht definiert. Gegenüber dem einfachen Verhältnis $v(h,s) = \frac{h-s}{h+s}$ hat das Ziehen der Wurzel den Vorteil, daß sich vereinzelte Ausreißer nicht so stark auf den Reputationswert auswirken. Abbildung 4.4 zeigt beispiel-

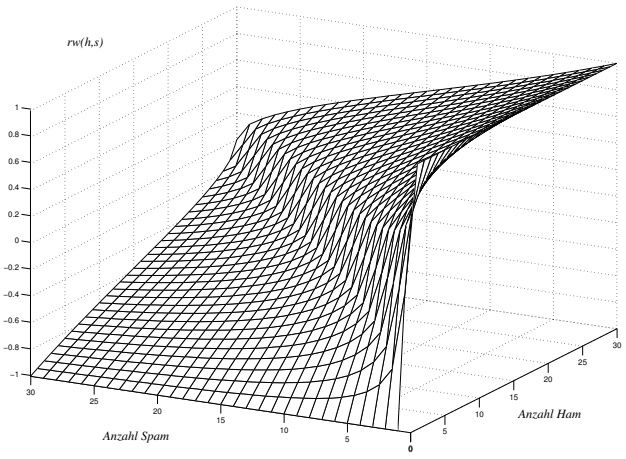

Abb. 4.3: Ratingfunktion *rw(h,s)*

haft den Verlauf des Reputationswertes für einen Mailversender, dessen erste zehn E-Mails als Ham gewertet wurden. Danach folgen abgesehen von einer kurzen Unterbrechung Spam-Mails. Zu Beginn des Spamversands fällt die Funktion *rw* langsamer als *v*, bewertet aber zunehmenden Spamversand dann stärker und reagiert auf die kurze Unterbrechung durch Ham weniger stark (Verbesserung des Spamwertes bei *rw* um 20%, bei *v* um 36%).

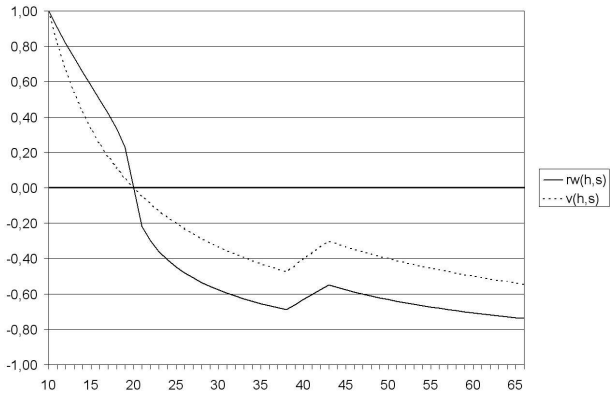

Abb. 4.4: Beispielhafte Entwicklung eines Reputationswertes

4.2.3.4 Akkreditierungsanfragen und -antworten

Akkreditierungsanfragen haben das Ziel, Reputationsinformationen über bislang unbekannte Mailversender einzuholen. Zum einen kann die Reputation einer Mailadresse, zum anderen die einer Domain erfragt werden. Eine Akkreditierungsanfrage beinhaltet folgende Informationen: die IP-Adresse des Absenders der Anfrage (entspricht der Adresse, an die Antworten gesendet werden), die Mailadresse bzw. der Domainname, dessen Reputation erfragt wird, ein Time-To-Live-Feld (TTL), welches die Rekursionstiefe festlegt, und eine Anfrage-ID, welche als Alleinstellungsmerkmal die Anfrage kennzeichnet und dazu dient, Zirkelschlüsse zu verhindern.

Eine Akkreditierungsantwort enthält auf jeden Fall neben der Anfrage-ID einen Domainnamen und dessen Reputationswert (bzw. „RW nicht gefunden“). Falls die Reputation einer Mailadresse erfragt wurde, enthält die Antwort zusätzlich die Mailadresse und deren RW.

4.2.3.5 Akkreditierungsadresse

Für die Mitteilung dieser Adresse bieten sich zwei Möglichkeiten an. Zum einen könnte die Adresse in einer speziellen Headerzeile zusammen mit einer E-Mail übertragen werden. Damit hat der Absender die Möglichkeit, den Server, auf dem sein Akkreditierungsdienst betrieben wird, relativ frei anzugeben. Die zweite Möglichkeit besteht darin, den Akkreditierungsdienst auf dem MTA des Absenders bzw. dessen Domain anzubieten. Dadurch kann die Adresse für Akkreditierungsanfragen aus den Absenderangaben ermittelt werden. Dieses Verfahren bietet den Vorteil eines etwas geringeren Verwaltungsaufwands und der Möglichkeit, auf bereits existierende Infrastruktur zurückgreifen zu können, ist aber bei der Verwendung unterschiedlicher Adressen für ein- und ausgehende Nachrichten etwas problematisch. Verwendet beispielsweise ein Benutzer A für eingehende Nachrichten `in@domain.tld` und für ausgehende `out@dom.tld`, würden Nutzer, welche A als Akkreditierungspartner nutzen wollen, die Anfragen an den Akkreditierungsdienst auf `dom.tld` richten, während A die Rückmeldungen für seine eingehenden Nachrichten an den Akkreditierungsdienst auf `domain.tld` sendet.

4.2.3.6 Rückmeldung des Empfängers

Wie bereits erwähnt wurde, ist eine Rückmeldung des Empfängers an den Akkreditierungsdienst essentiell. Durch eine enge Koppelung zwischen Akkreditierungsdienst (AD) und empfangenden MTA (siehe vorheriger Abschnitt) könnte das Fälschen von Reputationsprofilen erschwert werden, indem der Empfänger in seiner Rückmeldung lediglich die entsprechende E-Mail identifiziert (z. B. durch einen Hashwert) und seine Klassifizierung angibt. Der AD müßte nur überprüfen, ob der Absender der Rückmeldung mit dem tatsächlichen Empfänger übereinstimmt, und könnte dann aus der E-Mail den Absender extrahieren und den Reputationswert gemäß der Klassifikation des Benutzers anpassen. Diese Vorgehensweise bietet sich vor allem für ein vorhandenes Webinterface eines ESP an, bei denen ein „Mail als Spam behandeln"-Button schon vielfach vorhanden ist. In der überwiegenden Zahl der Fälle, wo die Nachrichten anhand eines Mailprogramms empfangen und gelesen werden, ist die Situation deutlich komplexer. Für diesen Einsatzzweck muß ein Protokoll entwickelt werden, anhand dessen entsprechende Rückmeldungen an den Akkreditierungsdienst geschickt werden können, und welches Mailprogrammen die Bereitstellung dieser Funktionalität erleichtert.

4.2.4 Der Akkreditierungsvorgang

Der Akkreditierungsvorgang ist der zentrale Bestandteil, welcher für den Datenaustausch des verteilten Reputationssystems zuständig ist. Der Abfragemechanismus funktioniert ähnlich wie ein DNS-Lookup, abgesehen vom Fehlen der hierarchischen Struktur. Dies bedeutet, daß es keine Root-Server gibt, welche den gesamten Namensraum – hier Adreßraum – verwalten, sondern die zentrale Instanz, aus der Adressen für Akkreditierungsanfragen bezogen werden, die Primärdatenbank eines Benutzers ist.

Für eine Anfrage wird eine sortierte Liste mit bevorzugten Akkreditierungspartnern erstellt; dies sind jene, welche in der Primärdatenbank einen möglichst guten Reputationswert aufweisen. Beginnend bei dem AP mit dem höchsten RW, werden rekursive Akkreditierungsanfragen verschickt. In diesem Zusammenhang bedeutet rekursiv, daß der AP seinerseits Akkreditie-

rungsanfragen an die von ihm bevorzugten APs schickt, falls er in seinem eigenen Datenbestand keine Antwort finden kann. Für diese Akkreditierungsanfragen wird keine neue Anfrage-ID generiert, sondern jene aus der ursprünglichen Anfrage übernommen, um Zirkelschlüsse zu erkennen und durch „RW nicht gefunden"-Antworten zu terminieren. In diesem Zusammenhang wird auch das TTL-Feld wichtig, welches um eins reduziert wird.

Prinzipiell sind auch iterative Anfragen denkbar, bei denen ein AP nicht selbst Anfragen stellt, sondern als Antwort seine bevorzugten Akkreditierungspartner mitteilt. Allerdings hat dieser Anfragetypus den Nachteil, daß dadurch unnötig Informationen preisgegeben werden. Dem Ursprung der Anfragen wird dadurch bekannt, welcher Partner bei wem einen guten Reputationswert hat oder, anders ausgedrückt, wer ein guter Freund von wem ist.

Zur Interpretation der gelieferten Ergebnisse wird die Bildung des Mittelwertes vorgeschlagen. Denkbar sind aber auch komplexere Verfahren, welche beispielsweise die Reputation des Auskunft gebenden Akkreditierungspartners mit einbeziehen.

Eine Möglichkeit, den entstehenden Datenverkehr zu begrenzen, ergibt sich durch die Zwischenspeicherung (Caching) der zurückgelieferte Ergebnisse an allen am Lookup beteiligten Knoten. Da eine Antwort über den gleichen Weg wie die Anfrage zum Ursprung zurückgelangt, können die Zwischenstationen das Ergebnis jeweils speichern und später für gleichlautende Anfragen wiederverwenden. Allerdings stellt sich hier wiederum die Frage, wie mehrere Antworten zu interpretieren sind und das Problem veralteter Daten im Cache (siehe auch 4.2.3.2).

4.2.5 Fehlende Authentifizierungsmerkmale

Falls keine Authentifizierungsmerkmale vorhanden sind, ist es schwierig, den tatsächlichen Absender zweifelfsfrei zu ermitteln. Hierbei sind zwei Möglichkeiten denkbar. Im ersten, einfacheren Fall handelt es sich um eine vom Absender neu erstellte Nachricht oder um ein Forwarding. Hierbei enthält der From-Header die Adresse des Absenders.

Im zweiten Fall handelt es sich um ein Resending. In diesem Fall bleibt die ursprüngliche Absenderadresse im From-Header erhalten. Statt dessen werden die sog. Resent-Header eingefügt, darunter auch das Resent-From-Feld, welches jene Adresse beinhaltet, von der aus die Nachricht erneut verschickt wurde. Da eine Nachricht mehrfach umgeleitet werden kann, können auch mehrere Resent-Header auftreten. In diesem Fall ist darauf zu achten, den letzten (am weitesten oben gelegenen) Header zu verwenden. Laut RFC 2822 ist jedoch die Verwendung der Resent-Header nicht verpflichtend: „Resent fields SHOULD be added to any message that is reintroduced by a user into the transport system.“[75]. Wird eine Nachricht umgeleitet, ohne Resent-Header zu verwenden, kann dadurch für diese E-Mail die (gute) Reputation des ursprünglichen Absenders erschlichen werden.

Aus diesen Gründen verbietet es sich, dieses Reputationssystem für nicht authentifizierte Absender zu verwenden. Es bleibt nichts anderes übrig, als den Spamwert als einziges Klassifizierungsmerkmal heranzuziehen und die E-Mail dementsprechend zu behandeln.

4.2.6 Problemanalyse

Durch Beantwortung einer Akkreditierungsanfrage und Auswertung des TTL-Feldes könnte ein Akkreditierungsdienst Wissen darüber erlangen, von welcher Adresse bzw. Domain eine E-Mail verschickt wurde, und eventuell auch den Adressaten erfahren. Dieses Wissen könnte brisant sein, wenn beispielsweise bekannt würde, daß konkurrierende Firmen miteinander kommunizieren. Allerdings wird die E-Mailadresse des Mailempfängers nicht publik. Jedoch könnte in einigen Fällen durch die IP-Adresse die zugehörige Domäne festgestellt werden, in wenigen Ausnahmefällen eventuell sogar die Identität des Empfängers. Allerdings sind solche Aussagen mit einer gewissen Unsicherheit verbunden, da der antwortende Akkreditierungsdienst selten mit Bestimmheit sagen kann, ob die Anfrage von einem anderen AD stammt oder vom Empfänger der E-Mail selbst. Dieses Problem kann entschärft werden, indem der Wert des TTL-Feldes für jede Anfrage innerhalb gewisser – für den

[75] [82] S.26 in Verbindung mit [7].

Akkreditierungsdienst unbekannter – Grenzen zufällig festgelegt wird. Eine weitere Möglichkeit wäre, nur Akkreditierungspartner der eigenen Domäne zu verwenden.[76]

Die gleiche Problematik liegt vor, wenn ein Akkreditierungsdienst eine Anfrage weiterleitet und hierbei das TTL-Feld den Wert eins aufweist und die Anfrage einen Reputationswert liefert. Allerdings kann nicht festgestellt werden, ob das Ergebnis aus der Caching- oder Primärdatenbank stammt. Dieses Problem kann noch weiter entschärft werden, indem gefordert wird, daß Anfragen mit TTL-Wert eins nur anhand der Caching-Datenbank beantwortet werden dürfen.

Spamversender könnten vermuten, daß Mailadressen mit einer guten Reputation zu Postfächern gehören, welche selbst häufig Mail erhalten und regelmäßig überprüft werden, und daher bevorzugt an solche Adressen Spam senden. Allerdings würde dann gerade diese Annahme sehr schnell zu einer schlechten Reputation des Spamversenders führen. Das Bekanntwerden einer guten oder schlechten Reputation ist Ziel dieses Verfahrens.

Darüber hinaus stellt ein Akkreditierungsdienst im Grunde eine Datenbank für Mailadressen dar und in Verbindung mit der Caching-Datenbank einen Dienst zum Sammeln von Adressen. Durch Akkreditierungsanfragen mit zufällig generierten Mailadressen kann deren Existenz durch Ausprobieren überprüft werden. Dies und die Auswertung der Caching-Datenbank des eigenen Akkreditierungsdienstes sind jedoch Möglichkeiten, welche in ähnlicher Form von Spamversendern bereits genutzt werden (Agenten zum Sammeln von Mailadressen bzw. Ausprobieren generierter Mailadressen und Auswertung des SMTP-Dialogs).

Am schwierigsten zu beantworten ist die Frage, inwiefern dieses Reputationssystem durch künstliche negative oder positive Reputationswerte sabotiert werden kann. Zum einen wird dieser falsche Wert in den Caching-Datenbanken aller beteiligten Akkreditierungsdienste gespeichert, zum anderen wird er an den Ursprung der Anfrage weitergeleitet, wo er mit dem Spamwert der Nachricht verglichen wird, was bei einer Diskrepanz zu ei-

[76] Ein zusätzlicher Vorteile wäre hierbei, daß deren Auskünften unter Umständen ein größeres Vertrauen entgegengebracht werden kann.

ner Interaktion des Empfängers führt. Für eine erfolgreich zugestellte Spam-Nachricht muß ein Spammer folgende drei Ziele erreichen:

1. Er muß bei anderen (möglichst vielen) Akkreditierungsdiensten einen guten Reputationswert haben.

2. Er muß in der Primärdatenbank des geplanten Empfängers der Spam-Mail einen guten Reputationswert haben, d. h. er muß wenigstens eine Nachricht an diesen verschickt haben, welche nicht als Spam gewertet wurde.

3. Der geplante Empfänger der Spam-Mail muß bei dieser ersten E-Mail einen der Akkreditierungsdienste abgefragt haben, bei denen der Spamversender eine gute Reputation hat.

Soweit momentan abzusehen ist, ist der zu betreibende Aufwand recht hoch bzw. nicht alle Ziele in Gänze durch den Spammer beeinflußbar (Punkt 3). Allerdings könnten sich die gefälschten Informationen im Cache stärker auswirken als die selbstreinigenden Kräfte, welche wirken, wenn eine Mailadresse aus dem Cache gelöscht wird und dafür ein entsprechender Eintrag in der Primärdatenbank angelegt wird (siehe 4.2.3.2). Eventuell ist an diesem Punkt eine Änderung der Caching-Strategie notwendig, z. B. durch Einführen eines Zeitstempels und Löschen veralteter Einträge.

4.2.7 Simulation und Ergebnisse

Um das vorgestellte Verfahren zu testen wurde unter Verwendung des PeerSim-Frameworks[77] eine Simulation implementiert. Hierbei ist zu beachten, daß E-Mail-Netzwerke besondere topologische Eigenschaften aufweisen. Wie in [20] gezeigt wurde, sind sie skalenfrei und zeigen gleichzeitig eine Kleine-Welt-Charakteristik. Das in der Simulation verwendete Netzwerk wurde daraufhin ausgelegt, beide Eigenschaften zu erfüllen: Die Verteilung des Grades g eines Knotens folgt dem Potenzgesetz $n(g) \propto g^{-1,94}$ und der durchschnittliche Grad liegt bei 3,98. Der Clustering-Koeffizient C_ν eines Knotens ν ist

[77] Siehe [49].

der Quotient aus der Anzahl der tatsächlichen Kanten (K) und der Anzahl aller möglichen Kanten ($\frac{1}{2}g_\nu(g_\nu - 1)$) zwischen den direkten Nachbarn von ν:

$$C_\nu = \frac{2K}{g_\nu(g_\nu - 1)}$$

Der Clustering-Koeffizient des Graphen ergibt sich dann als Mittelwert über die Koeffizienten aller Knoten. Für die Simulation beträgt der Wert 0,0328.

Zur Initialisierung der Simulation wird zunächst die Topologie gemäß obiger Vorgaben erstellt und eine feste Anzahl der Knoten zufällig ausgewählt und als „unbekannt“ markiert. Diese Knoten versenden während der Simulation Ham. Anschließend werden dem Netzwerk einige Knoten hinzugefügt, welche mit allen anderen Knoten verbunden und ebenfalls als „unbekannt“ markiert werden. Diese fungieren als Spam-Versender. Zur Initialisierung der primären Reputationsdatenbanken sendet jeder Knoten Nachrichten an seine Nachbarn, wobei keine Nachrichten an unbekannte Knoten oder von unbekannten Knoten verschickt werden. Zu Beginn der Simulation besteht das Netzwerk also einerseits aus Knoten, welche einen Teil ihrer Nachbarn kennen und somit über Akkreditierungspartner verfügen, und andererseits aus Knoten, für die im Netz noch keine Reputation gespeichert ist und anhand derer getestet werden kann, wie gut das Verfahren für die Klassifizierung von Nachrichten mit unbekannten Absendern geeignet ist (siehe Abbildung 4.5).

Die Simulation läuft zyklenweise ab. Jeder Zyklus besteht aus den Phasen Mailversand („unbekannte“ Knoten verschicken Nachrichten) und Klassifizierung (Überprüfung der Reputation). In der ersten Phase versendet jeder als „unbekannt“ markierte Knoten eine Nachricht an einen oder mehrere Nachbarn. Die Häufigkeit, wie oft ein Spam-Knoten eine Nachricht an den gleichen Empfänger senden darf, ist hierbei begrenzt, um den Ergebnissen aus Kapitel 4.2.2 (siehe Tabelle 4.2) gerecht zu werden. In der zweiten Phase wird gemäß dem in 4.2.1 bzw. 4.2.4 vorgestellten Verfahren der Spamwert und der Reputationswert ermittelt. Diese sind durch Flags vereinfacht dargestellt (gute/schlechte Reputation bzw. Spam-/Ham-Mail). Auf differenziertere Entscheidungsmöglichkeiten, welche beispielsweise durch die Verwendung von Gleitkommazahlen möglich wären, wurde verzichtet.

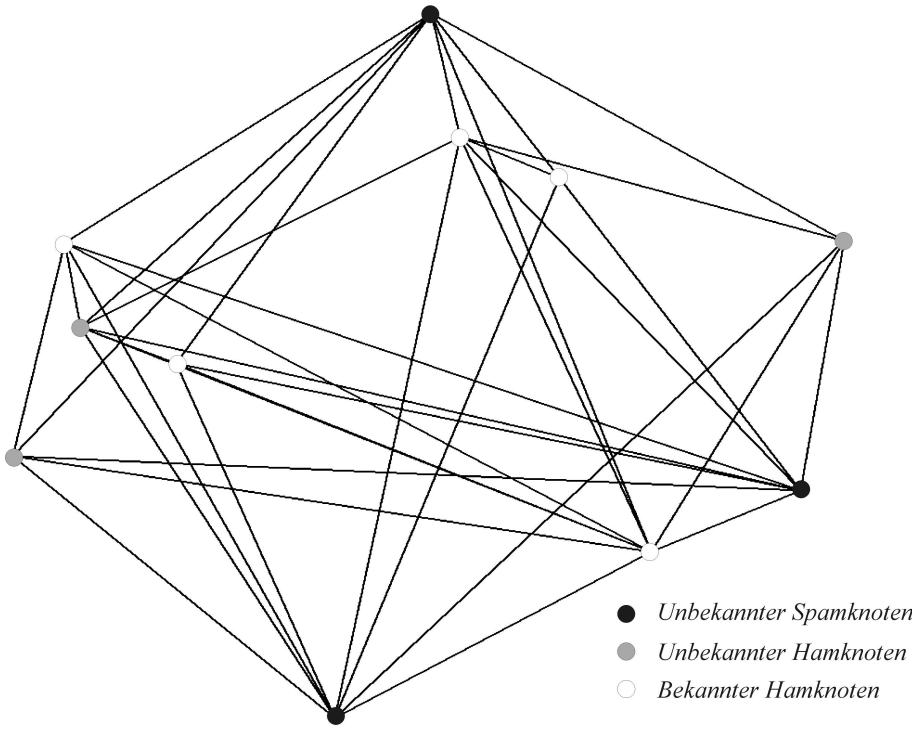

Abb. 4.5: Topologie des Simulationsnetzwerks (schematisch)

Das vorgestellte Verfahren sieht in zwei Fällen die Interaktion des Benutzers vor: Im ersten Fall wurde eine Reputation gefunden, aber der Reputationswert und der Spamwert stehen im Widerspruch. Im zweiten Fall konnte keine Reputation für den Absender ermittelt werden. Diese beiden Fälle werden in der Simulation wie folgt gehandhabt: Wenn keine Reputation gefunden wurde, wird die Mail lediglich aufgrund des Spamwertes klassifiziert. Ist der ermittelte Reputationswert zum Spamwert widersprüchlich, gibt der Reputationswert den Ausschlag für die Klassifizierung der Nachricht, wobei zusätzlich ermittelt wird, wie groß der durch den Bayesischen Filter induzierte Fehler ist.

Zunächst wurde getestet, welchen Einfluß die Parameter Suchtiefe (TTL) und Suchbreite (Anzahl der durch einen Knoten gestarteten Akkreditierungsanfragen für *eine* Reputationsauskunft) auf die Anzahl der gefundenen Reputationen haben. Wie Abbildung 4.6 zu entnehmen ist, verbessert sich die Anzahl der gefundenen Reputationen ab einer Suchtiefe von drei nicht mehr

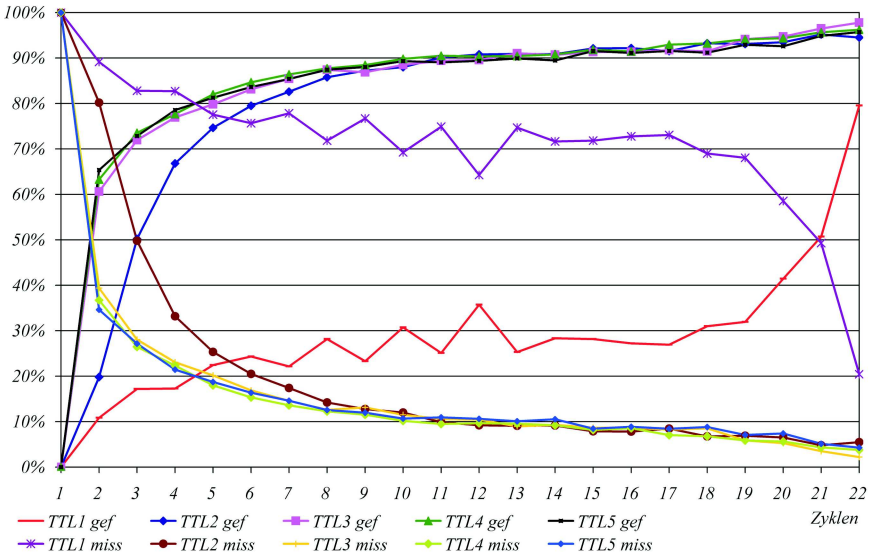

Abb. 4.6: Erfolgreiche und erfolglose Reputationsanfragen in Relation zum Mailaufkommen in Abhängigkeit von der Suchtiefe

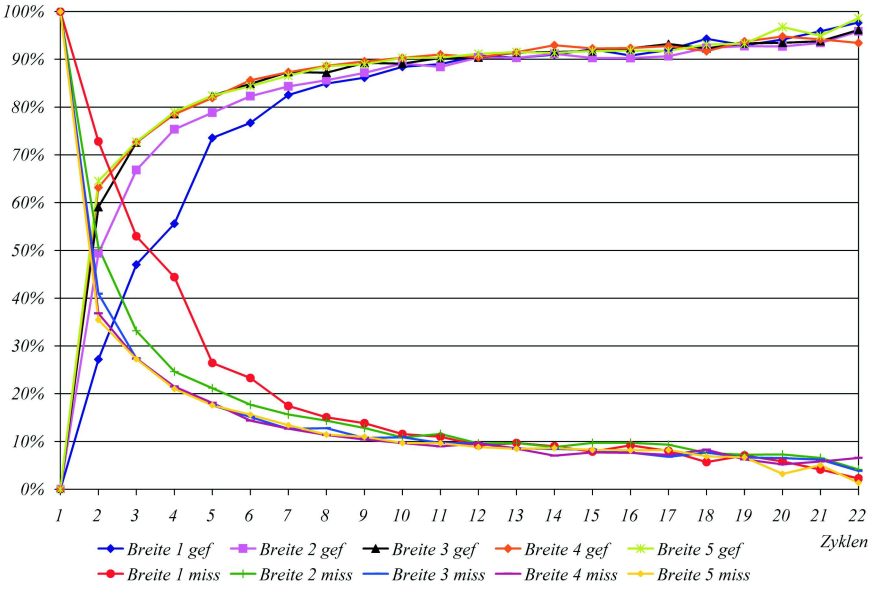

Abb. 4.7: Erfolgreiche und erfolglose Reputationsanfragen in Relation zum Mailaufkommen in Abhängigkeit von der Suchbreite

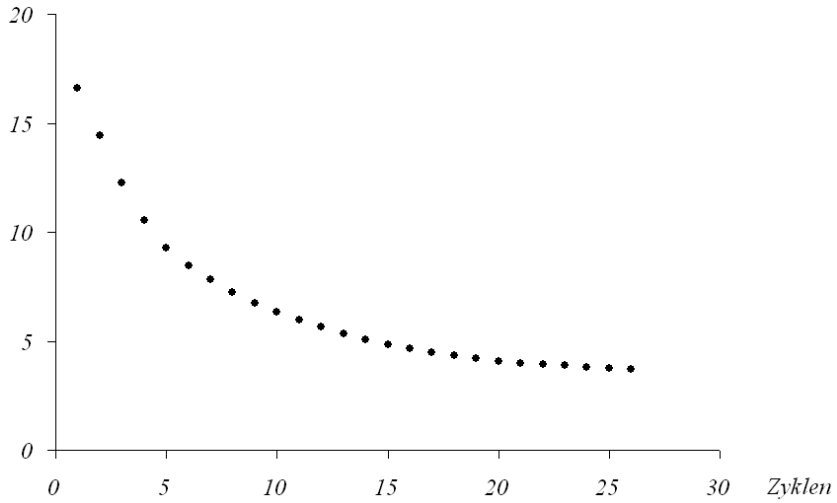

Abb. 4.8: Durchschnittliche Anzahl der Lookups pro Mail

nennenswert. Ähnliches gilt für die Suchbreite, bei der mit einem Wert von zwei bereits gute Ergebnisse erzielt werden, ein größerer Wert als drei bringt keine Verbesserung mehr (siehe Abbildung 4.7). Dies steht im Einklang mit der Topologie des Netzwerks, da der mittlere Grad eines Knotens, wie bereits erwähnt, bei 3,98 liegt.

Bei einer Suchtiefe und -breite von jeweils drei werden im Mittel pro E-Mail 6,6 Lookups durchgeführt. Abbildung 4.8 gibt einen besseren Überblick über die Entwicklung der Anzahl der durchgeführten Lookups pro Mail bei steigender Zyklenzahl.

Die Erkennungsrate des Verfahrens kann durch die Simulation nicht bestimmt werden, da die Benutzerinteraktion durch einen Bayesischen Filter simuliert wird. Allerdings ist eine Abschätzung in Abhängigkeit vom Fehler des Bayesischen Filter möglich. Für die Simulation wurden folgende, für einen Bayesischen Filter charakteristische Fehlerraten angenommen: Die Wahrscheinlichkeit für eine falsch postive Klassifizierung beträgt 4,7%, für eine falsch negative 0,3%. Abbildung 4.9 zeigt folgende Fehlerraten, wobei die Fehler für falsch Positive und falsch Negative jeweils addiert wurden: *Bayes* bezeichnet den Fehler, der durch einen Bayesischen Filter gemacht worden wäre und liegt wie zu erwarten im Bereich von 5%. *Rep miss* bezeichnet den Fehler, welcher bei E-Mails gemacht wurde, für deren Absender keine Repu-

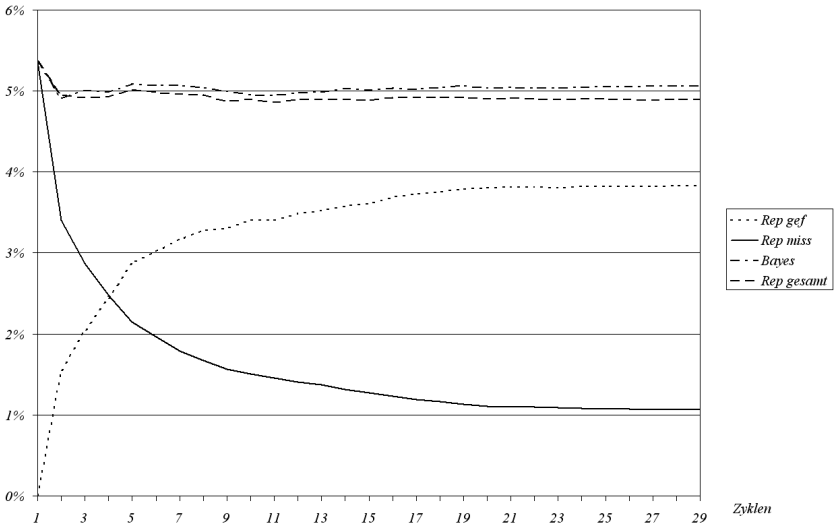

Abb. 4.9: Fehlerraten

tation ermittelt werden konnte und *Rep gef* jenen Fehler, der bei E-Mails gemacht wurde, für deren Absender zwar eine Reputation gefunden wurde, diese aber im Widerspruch zum Ergebnis des Bayesischen Filters steht. Dieser Fehler steigt mit der Zahl der gefundenen Reputationen zwangsläufig an. *Rep gesamt* schließlich, ist die Summe aus *Rep miss* und *Rep gef*. Würde im Falle eines Widerspruchs zwischen Spam- und Hamwert stets eine korrekte Klassifizierung durch den Benutzer erfolgen, würde *Rep gef* auf Null fallen, was wiederum zur Folge hätte, daß *Rep gesamt* mit *Rep miss* identisch wäre. Das bedeutet, daß der tatsächliche Fehler des Verfahrens zwischen *Rep gesamt* und *Rep miss* liegt, und damit geringer ist, als der eines Bayesischen Filters.

Be conservative in what you do, don't *be liberal in what you accept from others.*

in Anlehnung an John Postels Robustheitsprinzip

5 Zusammenfassung und Ausblick

5.1 Zusammenfassung

Spam, Werbemails, UCE, UBE, Phishing, Würmer, Joe Jobs – wie auch immer jegliche Formen unerwünschter E-Mails genannt werden mögen, sie stellen zur Zeit eines der irrationalsten Phänomene dar. Jeder Nutzer des Mediums E-Mail hat schon Spam-Mails erhalten und empfindet diese zumindest als Ärgernis, manchmal auch als deutliche Störung. Verschiedenen Analysen zufolge ist es nicht nur ein Ärgernis, sondern führt zu enormen betriebs- und volkswirtschaftlichen Kosten, weil Arbeitnehmer damit beschäftigt sind, die Stecknadeln aus dem Heu zu suchen, sprich normale Mails von Spam zu trennen, und Mailserver zum größten Teil unerwünschte Nachrichten transportieren. Um nicht allzu viele Spam-Mails zu erhalten, werden E-Mail-Adressen nur vorsichtig weitergegeben und nur unter Verwendung einfallsreicher Tricks veröffentlicht, damit sie nicht von einer Suchmaschine gefunden werden, anstatt sie als willkommene Möglichkeit zur unkomplizierten Kontaktaufnahme

anzubieten.

Trotz all dieser Einschränkungen und Hindernisse stellt die Kommunikation per E-Mail nach wie vor die wichtigste Internetanwendung dar. Es ist auch nicht davon auszugehen, daß sich in absehbarer Zeit etwas daran ändern wird. Seit langem stehen beispielsweise verschiedene Instant Messaging-Protokolle und -Anwendungen zur Verfügung, welche nicht nur viele Aufgaben übernehmen könnten, sondern oftmals sogar besser geeignet wären. Dennoch haben sie nicht annähernd den gleichen Stellenwert wie SMTP.

Da Spamming nicht zuletzt auch ein soziales Problem ist, wird es sich nicht allein durch technische Maßnahmen lösen lassen. Allerdings ist in den letzten Jahren die Sensibilität gegenüber der Problematik deutlich gewachsen, was vermehrt zu Gesetzesinitiativen und Urteilen geführt hat. Erste Fortschritte sind auch auf technischer Ebene festzustellen, und mit den in Kapitel 3 vorgestellten Protokollen SPF, Sender ID und DKIM sind Verfahren entstanden, welche sowohl den Bedürfnissen des Empfängers als auch jenen des Absenders entsprechen: Der Empfänger hat ein Interesse, die Identität des Absenders zu überprüfen, während der Absender verhindern möchte, daß seine Adresse unrechtmäßig benutzt wird. Da beide Seiten gleichermaßen von diesen Protokollen profitieren, ist davon auszugehen, daß sie in Zukunft verstärkt zum Einsatz kommen werden, was die Verwendung eines Reputationsverfahrens wie in Kapitel 4.2 beschrieben möglich macht. Falls der Absender einer Nachricht authentifiziert werden kann, ist es zulässig, hierfür eine Reputation zu speichern bzw. für weitere Entscheidungen Reputationsauskünfte zu Rate zu ziehen. Das vorgestellte verteilte Reputationsystem bietet hierbei bei vertretbarem Aufwand eine bessere Erkennungsrate, als der zugrundeliegende Bayesische Filter. Schlägt die Authentifizierung trotz vorhandener Informationen fehl, kann die Nachricht, je nach Richtlinien des Empfängers, abgewiesen oder gelöscht werden. Falls für eine Nachricht oder eine Domain keine Authentifizierungsmöglichkeiten gegeben sind, wird deren Transport und die Zustellung durch das vorgeschlagene Verfahren nicht erschwert.

Damit diese Verfahren ihre Wirksamkeit möglichst optimal entfalten können, wäre es wünschenswert, daß folgende Forderungen umgesetzt werden:

1. Von Domaininhabern sollte kontrolliert werden, auf welchen Wegen Nachrichten die Domain verlassen (dürfen) und für diese den externen Mailversand autorisieren.

2. Zum eigenen Schutz sollten Domaininhaber die erwähnten Authentifizierungsverfahren einsetzen und im Falle von DKIM alle ausgehenden Nachrichten signieren.

3. Domaininhaber sollten die Authentifizierung ihrer Benutzer verlangen, um dem Mißbrauch anderer Mailadressen der Domain vorzubeugen (cross user forgery).

4. Domaininhaber sollten ihre Authentifzierungs- und Signierungspolitik publizieren, um Empfängern eine Entscheidungshilfe bei der Klassifizierung der Nachrichten zu geben.

5. Das Ergebnis eines Authentifizierungsversuches muß festgehalten werden und möglichst genaue Informationen über den Absender beinhalten (`Received-SPF-` bzw. `Authentication-Results`-Header).

6. Mailprogramme sollten die Ergebnisse der vorausgegangenen Überprüfungen dem Benutzer darstellen (Anzeige des authentifizierten Absenders, des signierten Bereiches und der Reputation, etc.).

7. Durch die IETF müssen eindeutige Regelungen erlassen werden, in denen die verschiedenen Weiterleitungsformen definiert sind und festgelegt wird, welche Envelope- und Headereinträge bei welcher Weiterleitung wie geändert werden müssen.

8. Mailempfänger sollten eingehende Nachrichten auf das Vorhandensein von Authentifizierungsinformationen prüfen, dem Absender eine Reputation zuweisen und die Ergebnisse Dritten zur Verfügung stellen.

Wenn diese Forderungen auf breiter Ebene erfüllt sind, könnte sich ein Paradigmenwechsel vollziehen, wonach zukünftig nicht mehr wahllos jede Nachricht angenommen wird, sondern nur noch solche, deren Absender überprüft werden konnte.

Es darf jedoch nicht außer Acht gelassen werden, welche gesellschaftlichen Konsequenzen sich durch eine Authentifizierung auf Userebene möglicherweise ergeben. Wenn die erwähnten Verfahren akzeptiert und auf breiter Basis eingesetzt werden, wird dadurch der anonyme Nachrichtenversand erschwert. In freien, demokratischen Staaten mag dies akzeptabel und vielleicht auch erwünscht sein, für Bürger totalitärer Regime bedeutet dies jedoch möglicherweise ein erhöhtes Risiko bei der Wahrnehmung ihres Grundrechts auf freie Meinungsäußerung.

5.2 Ausblick

In der vorgestellten Fassung basiert das Reputationssystem ausschließlich auf der Zählung der eingegangenen Ham- und Spam-Mails. Prinzipiell könnten aber auch andere Eigenschaften des Mailverkehrs herangezogen werden. Denkbar wäre hierbei nicht nur eine Analyse der eingehenden, sondern auch der ausgehenden Nachrichten. Wird z. B. eine Nachricht an eine Adresse verschickt, welche laut Datenbank ein eher schlechtes Renommee hat, könnte dies ein Indiz sein, daß die Kommunikation mit dieser Adresse erwünscht ist und daher eine Verbesserung der Reputation angezeigt sein. Als weiteres Merkmal könnte der zeitliche Verlauf die Reputation beeinflussen. So könnten beispielsweise sehr viele Spam-Mails innerhalb eines kurzen Zeitraums zu einer stärkeren Verschlechterung der Reputation führen, als wenn die gleiche Menge Nachrichten über eine längere Zeitspanne empfangen würde. Dieser Ansatz könnte realisiert werden, indem Reputationswerte – positive wie negative – betragsmäßig automatisch verringert würden, je länger vom zugehörigen Inhaber keine Nachrichten empfangen wurden.

Ein interessanter Aspekt wäre die Übernahme und Zusammenführung fremder Reputationsdaten. Besonders interessant ist dies für Domänen, welche erst mit dem Aufbau einer Reputationsdatenbank begonnen haben, aber möglichst schnell einen hohen Nutzwert erreichen wollen, oder für Domänen, welche eine besondere gegenseitige Vertrauensstellung genießen. Allerdings verstärkt dies die Gefahr, daß sich im Laufe der Zeit einige wenige zentrale Instanzen herauskristallisieren, welche einen Großteil der Reputationsaus-

künfte übernehmen, was gerade durch den hier vorgestellten Ansatz vermieden werden soll. Möglicherweise führt dies auch zu einer Verminderung der Aussagekraft der Auskünfte.

Spamversender profitieren vor allem von zwei Dingen: Durch die Eigenschaften von SMTP fällt es ihnen leicht, ihre wahre Identität verborgen zu halten, und aufgrund fehlender oder nicht ausreichender Gesetze ist eine Verurteilung oft nicht möglich, selbst wenn der Verursacher bekannt ist. Durch die vorgestellten Authentifizierungsverfahren und der sich hoffentlich ausweitenden Praxis, nicht-autorisierte Nachrichten abzuweisen, können Spammer nicht mehr im Verborgenen agieren. Deswegen ist es erforderlich, daß die Bemühungen auf technischer Seite durch den Ausbau rechtlicher Grundlagen unterstützt werden. Eine international harmonisierte Rechtsprechung mit empfindlichen Geldstrafen trägt dazu bei, daß Spamming weniger lukrativ ist und damit uninteressant wird.

Alle angesprochenen Maßnahmen können dazu führen, daß SMTP für aufwendige Angriffsversuche wie DNS- oder TCP-Spoofing interessanter wird, da der Empfänger naturgemäß einer authentifizierten Nachricht mehr Vertrauen entgegenbringt. Inwiefern dies wirklich zu einer Bedrohung wird, kann jedoch getrost abgewartet werden und sollte zum jetzigen Zeitpunkt kein Grund darstellen, das vorgestellte Verfahren abzulehnen.

Abschließend bleibt festzustellen, daß das von John Postel für TCP geforderte und gerne für andere Internetprotokolle ins Feld geführte Robustheitsprinzip für den Mailversand per SMTP keine guten Dienste leistet. Die Forderung für eine verläßlichere Kommunikation per E-Mail sollte deswegen lauten:

Be conservative in what you do,
don't be liberal in what you accept from others.

Anhang

A Tabellen

A.1 Analyse From-Header

Mail-aufkommen	Anzahl Adressen Ham	Anzahl Adressen Spam
1	2880	29216
2	875	669
3	406	22
4	284	21
5	200	1
6	145	2
7	113	1
8	83	1
9	62	0
10	68	2
11	42	1
12	52	2
13	39	0
14	29	1
15	17	1
16	27	0
17	18	2

Tab. A.1: Anzahl unterschiedlicher Absenderadressen, Forts.

Mail-aufkommen	Anzahl Adressen Ham	Anzahl Adressen Spam
18	17	0
19	14	1
20	14	0
21	7	1
22	10	0
23	8	0
24	9	1
25	7	1
26	9	1
27	8	0
28	10	0
29	6	0
30	4	1
31	6	0
32	4	0
33	6	0
34	3	0
35	3	0
36	6	0
37	2	0
38	4	0
39	7	0
40	5	0
41	1	0
42	3	0
43	2	1
44	1	0
45	2	0
46	1	0
47	4	0
48	5	0
49	4	0
50	1	0
51	1	0
52	2	0
53	3	0
54	3	0
55	3	0

Tab. A.1: Anzahl unterschiedlicher Absenderadressen, Forts.

Mail-aufkommen	Anzahl Adressen Ham	Anzahl Adressen Spam
56	4	0
57	1	0
58	1	0
59	2	0
60	3	0
62	2	0
63	4	0
64	1	0
65	2	0
66	2	0
67	2	0
68	2	0
69	2	0
71	1	0
72	2	0
73	3	0
76	2	0
77	2	0
79	3	0
80	3	0
81	1	0
85	1	0
86	2	0
87	1	0
88	1	0
89	1	0
91	3	0
92	1	0
93	2	0
95	2	0
97	1	0
98	1	0
100	1	0
102	1	0
104	2	0
105	2	0
110	1	0
114	1	0

Tab. A.1: Anzahl unterschiedlicher Absenderadressen, Forts.

Mail-aufkommen	Anzahl Adressen Ham	Anzahl Adressen Spam
116	3	0
117	1	0
118	1	0
119	0	1
121	1	0
124	1	0
125	1	0
127	1	0
129	1	0
130	1	0
134	1	0
135	1	0
144	1	0
151	1	0
154	1	0
158	1	0
159	1	0
161	2	0
162	1	0
167	2	0
168	1	0
172	1	0
174	1	0
177	1	0
184	1	0
186	1	0
191	1	0
206	1	0
215	1	0
217	2	0
243	2	0
245	1	0
256	0	1
261	1	0
267	1	0
283	1	0
289	1	0
309	1	0

Tab. A.1: Anzahl unterschiedlicher Absenderadressen, Forts.

Mail-aufkommen	Anzahl Adressen Ham	Anzahl Adressen Spam
311	1	0
313	1	0
359	1	0
391	1	0
402	1	0
431	1	0
482	1	0
600	1	0
702	1	0
715	1	0

Tab. A.1: Anzahl unterschiedlicher Absenderadressen, kategorisiert nach Anzahl eingegangener Mails. Gesamtansicht

Mail-aufkommen	Anzahl Domains Ham	Anzahl Domains Spam
1	1 483	10 824
2	484	1 673
3	243	605
4	163	327
5	107	176
6	94	98
7	78	79
8	56	55
9	42	54
10	38	39
11	28	20
12	26	27
13	20	13
14	21	13
15	17	18
16	16	6
17	13	5
18	14	4
19	9	2
20	7	8
21	8	4
22	4	5

Tab. A.2: Anzahl unterschiedlicher Absenderdomains, Forts.

Mail-aufkommen	Anzahl Domains Ham	Anzahl Domains Spam
23	11	1
24	8	4
25	5	2
26	5	3
27	6	2
28	6	1
29	5	4
30	3	1
31	3	1
32	4	2
33	8	3
34	1	4
35	1	3
36	2	2
37	3	4
38	3	3
39	5	2
40	2	5
41	2	1
42	3	1
43	2	5
44	2	2
45	1	3
46	0	1
47	4	1
48	1	1
49	1	1
50	2	1
51	2	2
52	2	0
53	3	1
54	4	6
55	1	2
56	1	2
57	1	2
58	2	1
59	3	1
60	1	0

Tab. A.2: Anzahl unterschiedlicher Absenderdomains, Forts.

Mail-aufkommen	Anzahl Domains Ham	Anzahl Domains Spam
62	1	0
63	3	1
65	2	3
66	3	0
67	1	1
68	1	0
69	2	1
71	2	2
72	2	0
73	4	0
74	1	1
77	2	0
78	0	1
79	1	0
80	2	1
84	0	1
85	1	0
86	1	0
89	2	0
93	2	0
95	1	0
97	1	0
98	1	0
100	1	0
102	1	0
103	1	0
104	2	0
107	0	1
108	1	0
111	0	1
113	1	2
115	1	0
116	1	0
117	1	0
124	1	0
125	1	0
128	1	0
130	2	0

Tab. A.2: Anzahl unterschiedlicher Absenderdomains, Forts.

Mail-aufkommen	Anzahl Domains Ham	Anzahl Domains Spam
134	1	0
135	1	0
152	1	0
158	1	0
159	1	0
161	2	0
162	1	0
167	1	0
168	1	0
172	1	0
174	2	0
175	0	1
177	1	0
178	0	1
186	1	0
191	1	0
214	1	0
227	0	1
246	1	0
254	1	0
258	0	1
261	1	0
267	1	0
287	0	1
288	1	0
289	1	0
313	1	0
321	1	0
331	0	1
333	1	0
342	0	1
363	1	0
405	1	0
432	1	0
443	1	0
463	1	0
468	1	0
516	1	0

Tab. A.2: Anzahl unterschiedlicher Absenderdomains, Forts.

Mail-aufkommen	Anzahl Domains Ham	Anzahl Domains Spam
537	1	0
598	1	0
702	1	0
715	1	0
794	1	0
819	1	0
884	1	0
1113	1	0
1246	0	1
1422	0	1
1943	1	0
3464	1	0

Tab. A.2: Anzahl unterschiedlicher Absenderdomains, kategorisiert nach Anzahl eingegangener Mails. Gesamtansicht

Verzeichnisse

Literaturverzeichnis

[1] Allman, Eric ; Callas, Jon ; Delany, Mark ; Libbey, Miles ; Fenton, Jim ; Thomas, Michael. *DomainKeys Identified Mail (DKIM).* URL: http://ietfreport.isoc.org/all-ids/draft-ietf-dkim-base-04.txt. Juli 2006

[2] Anonym: Ferngesteuerte Spam-Armeen. In: *c't – magazin für computer technik* (2004), Nr. 05, S. 18–22

[3] ASRG. *Anti-Spam Research Group.* URL: http://asrg.sp.am/. 2004

[4] Atkins, Derek ; Stallings, William ; Zimmermann, Philip: *PGP Message Exchange Formats.* URL: http://www.ietf.org/rfc/rfc1991.txt: IETF, August 1996

[5] Atkinson, Robert G. ; et al. *Reducing unwanted and unsolicited electronic messages by preventing connection hijacking and domain spoofing.* US Patent & Trademark Office; URL: http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&co1=AND&d=PG01&s1=684020.APN.&OS=APN/684020&RS=APN/684020. September 2004

[6] Back, Adam. *Hashcash - A Denial of Service Counter-Measure.* URL: http://www.hashcash.org/papers/hashcash.pdf. August 2002

[7] Bradner, Scott: *Key words for use in RFCs to Indicate Requirement Levels.* URL: http://www.ietf.org/rfc/rfc2119.txt: IETF, März 1997

[8] Brand, Raymond S. ; Sherzer, Laurence ; Rognlie, Richard W. *Designated Relays Inquiry Protocol (DRIP).* URL: http://www.sherzer.net/draft-brand-drip-02.txt. Oktober 2003

[9] Braun, Dietmar ; Kocovski, Jan ; Rickert, Thomas ; Dr. Waldhauser, Béla ; eco – Verband der deutschen Internetwirtschaft e.V. (Hrsg.). *White Paper der Anti Spam Task Force (Vollversion)*. URL: `http://www.eco.de/servlet/PB/menu/1446039_l1/index.html`. September 2004

[10] Callas, Jon ; Donnerhacke, Lutz ; Finney, Hal ; Thayer, Rodney: *OpenPGP Message Format*. URL: `http://www.ietf.org/rfc/rfc2440.txt`: IETF, November 1998

[11] CipherTrust. *TrustedSource E-mail Reputation System*. URL: `http://www.ciphertrust.com/products/trustedsource/`

[12] Cloudmark. *SpamNet*. URL: `http://www.cloudmark.com/`

[13] Damiani, Ernesto ; Vimercati, Sabrina De C. ; Paraboschi, Stefano ; Samarati, Pierangela: P2P-Based Collaborative Spam Detection and Filtering. In: *p2p* 00 (2004), S. 176–183. ISBN 0–7695–2156–8

[14] Danisch, Hadmut. *The RMX DNS RR and method for lightweight SMTP sender authorization*. URL: `http://www.danisch.de/work/security/txt/draft-danisch-dns-rr-smtp-04.txt`. Mai 2004

[15] Declude. *List of All Known DNS-based Spam Databases*. URL: `http://www.declude.com/Articles.asp?ID=97`

[16] Delany, Mark. *Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys)*. URL: `http://ietfreport.isoc.org/all-ids/draft-delany-domainkeys-base-02.txt`. März 2005

[17] DENIC. *Domainzahlenvergleich der zehn größten TLDs*. URL: `http://www.denic.de/de/domains/statistiken/domainvergleich_tlds/index.html`. Juli 2006

[18] Dr. Mash, Jørgen. *drbcheck: dr. Jørgen Mash's DNS database list checker*. URL: `http://moensted.dk/spam/`

[19] Eastelake, Donald E. ; Jones, Paul E.: *US Secure Hash Algorithm 1 (SHA1)*. URL: `http://www.ietf.org/rfc/rfc3174.txt`: IETF, September 2001

[20] Ebel, Holger ; Mielsch, Lutz-Ingo ; Bornholdt, Stefan: Scale-free topology of e-mail networks. In: *Phys. Rev. E* 66 (2002), Sep, Nr. 3, S. 035103

[21] Fecyk, Gordon. *Designated Mailers Protocol*. URL: http://www.pan-am.ca/dmp/draft-fecyk-dmp-02.txt. Mai 2004

[22] Feist, Sharael. *The father of modern spam speaks*. URL: http://news.com.com/2008-1082-868483.html. März 2002

[23] Fenton, Jim ; Thomas, Michael. *Identified Internet Mail*. URL: http://www.identifiedmail.com/draft-fenton-identified-mail.txt. Mai 2005

[24] Goodmail. *Goodmail Systems*. URL: http://www.goodmailsystems.com/

[25] Gray, Alan ; Haahr, Mads: Personalised, Collaborative Spam Filtering. In: *CEAS*, 2004. – URL: http://www.ceas.cc/papers-2004/132.pdf

[26] Green, David N. *Domain-Authorized SMTP Mail*. URL: http://www.ops.ietf.org/lists/namedroppers/namedroppers.2002/msg00656.html. Juni 2002

[27] Habeas. *Habeas is an email trust authority that certifies the practices of legitimate email senders*. URL: http://www.habeas.com/index.php

[28] Heise News. *GMX erneut auf Antispam-Blockliste*. URL: http://www.heise.de/newsticker/meldung/64798

[29] Heise News. *Spam-Blockliste lief Amok*. URL: http://www.heise.de/newsticker/meldung/65358

[30] Heise News. *Millionenstrafe für Spammer*. URL: http://www.heise.de/newsticker/meldung/41402. Oktober 2003

[31] Heise News. *Spam belastet Europas Unternehmen*. URL: http://www.heise.de/newsticker/meldung/33417. Januar 2003

[32] Heise News. *BSA: Spam-Mails gefährden regulären Online-Handel*. URL: http://www.heise.de/newsticker/meldung/54086. Dezember 2004

[33] Heise News. *Mehr als 40 Prozent Spam-Mails in Deutschland*. URL: http://www.heise.de/newsticker/meldung/48827. Juli 2004

[34] Heise News. *Spam: Gericht spricht Provider 1 Milliarde US-Dollar Entschädigung zu.* URL: http://www.heise.de/newsticker/meldung/54439. Dezember 2004

[35] Heise News. *Spammer muss Microsoft 4 Millionen US-Dollar Schadensersatz zahlen.* URL: http://www.heise.de/newsticker/meldung/49162. Juli 2004

[36] Heise News. *Botnetze an Spammer vermietet.* URL: http://www.heise.de/newsticker/meldung/65747. November 2005

[37] Heise News. *CipherTrust: Täglich 172.000 neue Spam-Zombies.* URL: http://www.heise.de/newsticker/meldung/59970. Mai 2005

[38] Heise News. *Microsoft bekommt 7 Millionen US-Dollar vom "Spam King".* URL: http://www.heise.de/newsticker/meldung/62610. August 2005

[39] Heise News. *24-jähriger Massen-Spammer zahlt eine Million US-Dollar Strafe.* URL: http://www.heise.de/newsticker/meldung/73914. Juni 2006

[40] Heise News. *Bounce-Mails werden zur Plage.* URL: http://www.heise.de/newsticker/meldung/73106. Mai 2006

[41] Heise News. *Spammer muss 11,2 Milliarden US-Dollar zahlen.* URL: http://www.heise.de/newsticker/meldung/68031. Januar 2006

[42] Heise News. *Spammer soll für mindestens zwei Jahre hinter Gitter.* URL: http://www.heise.de/newsticker/meldung/68320. Januar 2006

[43] Heise News. *T-Online verzeichnet eine Milliarde Spam-Mails pro Tag.* URL: http://www.heise.de/newsticker/meldung/72324. April 2006

[44] Heise News. *US-Behörden: Einigung in Verfahren gegen Spam-Versender.* URL: http://www.heise.de/newsticker/meldung/71842. April 2006

[45] Heise News. *US-Spammer zahlt 1,1 Millionen US-Dollar Strafe.* URL: http://www.heise.de/newsticker/meldung/70777. März 2006

[46] Hormel Foods. *SPAM and the Internet.* URL: http://www.spam.com/ci/ci_in.htm. 2003

[47] Internet Systems Consortium (ISC). *Distribution of Top-Level Domain Names by Host Count*. URL: `http://www.isc.org/ops/ds/reports/2006-07/dist-bynum.php`. Juli 2006

[48] Ironport. *The Leader in Network Security for Email*. URL: `http://www.ironport.com/`

[49] Jelasity, Márk ; Jesi, Gian P. ; Montresor, Alberto ; Voulgaris, Spyros. *PeerSim: A Peer-to-Peer Simulator*. URL: `http://peersim.sourceforge.net/`

[50] Johansson, Eric S. ; Dawson, Keith. *Camram*. URL: `http://www.camram.org/cr-files/camram.pdf`. Januar 2003

[51] Jung, Jason J. ; Jo, GeunSik: Collaborative Junk E-mail Filtering Based on Multi-agent Systems. In: Chung, Chin-Wan (Hrsg.) ; Kim, Chong kwon (Hrsg.) ; Kim, Won (Hrsg.) ; Ling, Tok W. (Hrsg.) ; Song, Kwan H. (Hrsg.): *Human.Society@Internet 2003* Bd. 2713, Springer, 2003. – ISBN 3–540–40456–2, S. 218–227

[52] Klensin, John C.: *Simple Mail Transfer Protocol*. URL: `http://www.ietf.org/rfc/rfc2821.txt`: IETF, April 2001

[53] Kucherawy, Murray S. *Message Header for Indicating Sender Authentication Status*. URL: `http://ietfreport.isoc.org/all-ids/draft-kucherawy-sender-auth-header-03.txt`. Februar 2006

[54] Lazzari, Lorenzo ; Mari, Marco ; Poggi, Agostino: CAFE – Collaborative Agents for Filtering E-mails. In: *wetice* 0 (2005), S. 356–361. – ISSN 1524–4547

[55] Levine, John R. *A Flexible Method to Validate SMTP Senders in DNS*. URL: `http://www.watersprings.org/pub/id/draft-levine-fsv-01.txt`. April 2004

[56] Lyon, Jim: *Purported Responsible Address in E-Mail Messages*. URL: `http://www.ietf.org/rfc/rfc4407.txt`: IETF, April 2006

[57] Lyon, Jim ; Wong, Meng W.: *Sender ID: Authenticating E-Mail*. URL: `http://www.ietf.org/rfc/rfc4406.txt`: IETF, April 2006

[58] MessageLabs. *Spam Intercepts*. URL: `http://www.messagelabs.co.uk/publishedcontent/publish/threat_watch_dotcom_en/threat_statistics/spam_intercepts/DA_114633.chp.html`

[59] MessageLabs. *MessageLabs Intelligence Report: Annual Report 2004*. URL: http://www.messagelabs.co.uk/portal/server.pt/gateway/PTARGS_0_0_456_323_-323_43/http%3B/0120-0176-CTC1%3B8080/publishedcontent/publish/_dotcom_libraries_en/files/monthly_reports/2004/messagelabs_intelligence_report__annual_report_2004_5.pdf. 2005

[60] MessageLabs. *MessageLabs Intelligence Report: Annual Report 2005*. URL: http://www.messagelabs.co.uk/portal/server.pt/gateway/PTARGS_0_0_389_594_-594_43/http%3B/0120-0176-CTC1%3B8080/publishedcontent/publish/_dotcom_libraries_en/files/monthly_reports/2005_annual_report_5.pdf. 2006

[61] Micro, Trend. *Trend Micro Network Reputation Services*. URL: http://www.trendmicro.com/en/products/nrs/overview.htm

[62] Microsoft Corporation. *Caller ID for E-Mail*. URL: http://download.microsoft.com/download/2/e/2/2e2850b8-2747-4394-a5a9-d06b5b9b1a4c/callerid_email.pdf

[63] Mockapetris, Paul: *Domain Names – Concepts and Facilities*. URL: http://www.ietf.org/rfc/rfc1034.txt: IETF, November 1987

[64] Mockapetris, Paul: *Domain Names – Implementation and Specification*. URL: http://www.ietf.org/rfc/rfc1035.txt: IETF, November 1987

[65] Monty Python: *Spam*. URL: http://www.intriguing.com/mp/_scripts/another.asp. 1970. – auch auf CD: „Another Record“. ASIN: B000000HQC. Label: ASIN

[66] Myers, John G.: *SMTP Service Extension for Authentication*. URL: http://www.ietf.org/rfc/rfc2554.txt: IETF, März 1999

[67] NN. *Homepage der IETF-Arbeitsgruppe DKIM*. URL: http://www.ietf.org/html.charters/dkim-charter.html

[68] NN. *Homepage DKIM*. URL: http://mipassoc.org/dkim/index.html

[69] NN. *JAP Anonymity & Privacy*. URL: http://anon.inf.tu-dresden.de/

[70] NN. *Kung PoW! Spam fighting*. URL: http://www.camram.org/zombielogic

[71] NN. *SORBS (Spam and Open-Relay Blocking System).* URL: `http://www.nl.sorbs.net/`

[72] NN. *Spamcop.net.* URL: `http://www.spamcop.net`

[73] NN. *Fact Sheet: President Bush Signs Anti-Spam Law.* URL: `http://www.whitehouse.gov/news/releases/2003/12/20031216-4.html`. Dezember 2003

[74] NN. *You Might Be An Anti-Spam Kook If...* URL: `http://www.rhyolite.com/anti-spam/you-might-be.html`. Oktober 2005

[75] NN. *Spam Laws.* URL: `http://www.spamlaws.com`. März 2006

[76] Partridge, Craig: *Mail Routing and the Domain System.* URL: `http://www.ietf.org/rfc/rfc974.txt`: IETF, Januar 1986

[77] Popov, Kirill ; McDonald, Loren. *Deliverability:A Challenge for 8 of 10 Marketers.* URL: `http://www.clickz.com/showPage.html?page=3597946`. April 2006

[78] Prakash, Vipul V. *Vipul's Razor.* URL: `http://razor.sourceforge.net`

[79] Prakash, Vipul V. ; O'Donnell, Adam: Fighting spam with reputation systems. In: *Queue* 3 (2005), Nr. 9, S. 36–41. – ISSN 1542–7730

[80] Ramsdell, Blake: *Secure/Multipurpose Internet Mail Extensions (S/-MIME) Version 3.1 Certificate Handling.* URL: `http://www.ietf.org/rfc/rfc3850.txt`: IETF, Juli 2004

[81] Ramsdell, Blake: *Secure/Multipurpose Internet Mail Extensions (S/-MIME) Version 3.1 Message Specification.* URL: `http://www.ietf.org/rfc/rfc3851.txt`: IETF, Juli 2004

[82] Resnick, Peter W.: *Internet Message Format.* URL: `http://www.ietf.org/rfc/rfc2822.txt`: IETF, April 2001

[83] ReturnPath. *Return Path Solutions for Increased Email Delivery, Performance.* URL: `http://www.returnpath.biz`

[84] Rhyolite Software. *Distributed Checksum Clearinghouse.* URL: `http://www.rhyolite.com/anti-spam/`

[85] Rhyolite Software. *Distributed Checksum Clearinghouse Graphs.* URL: http://www.rhyolite.com/anti-spam/dcc/graphs/. September 2006

[86] Sahami, Mehran ; Dumais, Susan ; Heckerman, David ; Horvitz, Eric: A Bayesian Approach to Filtering Junk E-Mail. In: *Learning for Text Categorization: Papers from the 1998 Workshop.* Madison, Wisconsin : AAAI Technical Report WS-98-05, 1998

[87] Schneier, Bruce: *Angewandte Kryptographie.* München : Pearson Studium, 2006

[88] Shevek. *The Sender Rewriting Scheme.* URL: http://www.libsrs2.org/srs/srs.pdf. Juni 2004

[89] Sophos. *Sophos outs 'dirty dozen' spam producing countries.* URL: http://www.sophos.com/pressoffice/news/articles/2004/02/pr_au_20040227dirtydozen.html. Februar 2004

[90] Sophos. *Sophos report reveals latest 'dirty dozen' spam relaying countries.* URL: http://www.sophos.com/pressoffice/news/articles/2006/04/dirtydozapr06.html. April 2006

[91] Spamhaus. *The Definition of Spam.* URL: http://www.spamhaus.org/definition.html

[92] Spamhaus. *The Spamhaus Project.* URL: http://www.spamhaus.org

[93] Stumpf, Markus ; Hoehne, Steff. *Marking Mail Transfer Agents in Reverse DNS with TXT RRs.* URL: http://mtamark.space.net/draft-stumpf-dns-mtamark-04.txt. Mai 2005

[94] Templeton, Brad. *Origin of the term "spam" to mean net abuse.* URL: http://www.templetons.com/brad/spamterm.html

[95] Templeton, Brad. *Reaction to the DEC Spam of 1978.* URL: http://www.templetons.com/brad/spamreact.html

[96] TrustedSource. *Domains with DomainKeys/DKIM Information.* URL: http://www.trustedsource.org/dkim.php

[97] TrustedSource. *Domains with SPF/SenderID Information.* URL: http://www.trustedsource.org/senderid.php

[98] TrustedSource. *TrustedSource Portal.* URL: http://www.trustedsource.org/

[99] U.S. Department of Commerce ; National Institute of Standards and Technology ; Information Technology Laboratory. *Secure Hash Signature Standard.* URL: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf. August 2002

[100] Vixie, Paul. *Repudiating MAIL FROM.* URL: http://www.ops.ietf.org/lists/namedroppers/namedroppers.2002/msg00658.html. Mai 2002

[101] Weinman, Bill. *AMTP – a replacement for SMTP.* URL: http://amtp.bw.org/

[102] Wikipedia. *Breidbart-Index.* URL: http://de.wikipedia.org/wiki/Breidbart-Index. Februar 2005

[103] Wired News. *Swollen Orders Show Spam's Allure.* URL: http://www.wired.com/news/business/0,1367,59907,00.html. August 2003

[104] Wong, Meng W. ; Schlitt, Wayne: *Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1.* 1. URL: http://www.ietf.org/rfc/rfc4408.txt: IETF, April 2006

[105] Zenger, Rejo. *confession for two: a spammer spills it all.* URL: https://rejo.zenger.nl/abuse/1085493870.php. Dezember 2005

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Codebeispiele

Abkürzungen und Akronyme

bzw.	beziehungsweise
d. h.	das heißt
etc.	et cetera
ggf.	gegebenenfalls
Mio.	Millionen
Mrd.	Milliarden
sog.	sogenannt
s. u.	siehe unten
usw.	und so weiter
z. B.	zum Beispiel

ABNF	Augmented Backus–Naur Form
AD	Akkreditierungsdienst
AGB	Allgemeine Geschäftsbedingungen
AMTP	Authenticated Mail Transfer Protocol
AP	Akkreditierungspartner
ASRG	Anti Spam Research Group
CIDR	Classless Inter-Domain Routing
CR	Challenge/Response
CRLF	Carriage Return und Line Feed (Zeilenwechsel)
DCC	Distributed Checksum Clearinghouse
DDN	Dotted Decimal Notation
DDOS	Distributed Denial of Service
DK	DomainKeys
DKIM	DomainKeys Identified Mail
DMP	Designated Mailers Protocol
DNS	Domain Name System
DNSBL	DNS-based Blackhole List
DRIP	Designated Relays Inquiry Protocol
DSN	Delivery Status Notification
ESP	E-Mail Service Provider

FQDN	Full Qualified Domain Name
FSV	Flexible Sender Validation
FWS	Folding Whitespace
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IIM	Identified Internet Mail
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IRTF	Internet Research Task Force
ISP	Internet Service Provider
JAP	Java Anon Proxy
KRS	Key Registration Server
MARID	MTA Authorization Records in DNS
MDA	Mail Delivery Agent
MIME	Multipurpose Internet Mail Extensions
MSA	Mail Submission Agent
MTA	Mail Transfer Agent
MTAmark	Marking MTAs in Reverse DNS
MUA	Mail User Agent
MX	Mail eXchange
OSI	Open Systems Interconnection
PDA	Personal Digital Assistant
PGP	Pretty Good Privacy
POP3	Post Office Protocol Version 3
PRA	Purported Responsible Address
RBL	Realtime Blackhole List
RFC	Request for Comment
RMX	Reverse MX RR
RP	Responsible Person
RR	Resource Record
RSA	asymmetrisches Verschlüsselungsverfahren; nach den Autoren Rivest, Shamir und Adleman benannt.
RSP	Reputation Service Provider
RW	Reputationswert
SHA1	US Secure Hash Algorithm 1; beschrieben in [19]
SHA256	ein Hashalgorithmus; Bestandteil des Secure Hash Standard, siehe [99]
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMS	Short Message Service

SMTP	Simple Mail Transfer Protocol
SMTP-AUTH	Authentifizierung gemäß RFC 2554 [66]
SPF	Sender Policy Framework
SRS	Sender Rewriting Scheme
SW	Spamwert; Wahrscheinlichkeit, ob eine Nachricht Spam ist
TCP	Transport Control Protocol
TTL	Time To Live
UBE	Unsolicited Bulk Email
UCE	Unsolicited Commercial Email
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
UTC	Temps Universel Coordonné
UWG	Gesetz gegen den unlauteren Wettbewerb
WWW	World Wide Weg
XML	eXtended Markup Language

Danksagung

An dieser Stelle möchte ich meinem Doktorvater Prof. Dr. Manfred Sommer für die freie Wahl des Themengebiets und die Möglichkeit zur unabhängigen Arbeit danken. Sowohl fachlich und organisatorisch, als auch menschlich war es stets ein sehr angenehmes Arbeitsverhältnis.

Herrn Prof. Dr. Bernd Freisleben danke ich für die zahlreichen Hinweise und Anregungen.

Ein besonderer Dank gebührt meinen Kollegen David Kämpf und Markus Vincon, welche mir gegen Ende viel Arbeit bei der Vorlesungsbetreuung abgenommen haben und auch sonst mit Rat und Tat zur Seite standen. Des weiteren seien an dieser Stelle Jost Berthold, Barbara Krzensk, Steffen Priebe und Dr. Axel Schröder erwähnt, welche stets für Fragen zur Verfügung standen und mir ihrerseits mit kritischen Anmerkungen geholfen haben.

Vielen Dank auch an Lydia Heinbächer, Frau Walldorf und Barbara Dinklage für all die großen und kleinen Hilfen während meiner Anstellung am Fachbereich.

Ohne die Hilfe meiner Schwester Monika Meisl wäre diese Arbeit sicherlich um einige Rechtschreib- und Grammatikfehler reicher. Vielen Dank!

Der größte Dank gilt meiner Familie und insbesondere meiner Frau, welche mir stets den Rücken freigehalten und die nötige Zeit für diese Arbeit gegeben hat.

Erklärung des Verfassers

Ich versichere, dass ich meine Dissertation

Ein verteiltes Reputationssystem zur Filterung
unerwünschter E-Mails

selbständig und ohne unerlaubte Hilfe angefertigt und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen und Hilfen bedient habe.

Die Dissertation wurde in der jetzigen oder einer ähnlichen Form weder bei einer anderen Hochschule eingereicht noch hat sie sonstigen Prüfungszwecken gedient.

Marburg, im Oktober 2006

Lebenslauf

Roman Meisl
Odenwaldstr. 42
35043 Marburg

geboren am 19. März 1971 in München
Eltern: Hans Meisl und Theresia Meisl, geb. Schenk

Schulbildung

1977-1981	Adalbert-Stifter-Grundschule in Kaufbeuren
1981-1983	Gustav-Leutelt-Hauptschule in Kaufbeuren
1983-1987	Staatliche Realschule Kaufbeuren
1987-1991	Staatliches Gymnasium Kaufbeuren

Zivildienst

1991-1992	Individuelle Schwerstbehinderten-Betreuung bei der Katholisch-Evangelischen Sozialstation Kaufbeuren

Akademische Ausbildung

1992-2001	Studium der Mathematik mit Nebenfach Psychologie an der Philipps-Universität Marburg
2001-2006	Anstellung als wissenschaftlicher Mitarbeiter am Fachbereich Mathematik und Informatik an der Philipps-Universität Marburg